

Treball de Fi de Grau

Grau en Enginyeria en Tecnologies Industrials

Creació d'un qüestionari gamificat

MEMÒRIA

Autor: Emma Villanueva Rourera
Director: Daniela Tost
Convocatòria: Setembre 2019



Escola Tècnica Superior
d'Enginyeria Industrial de Barcelona



Resum

En un món cada cop més tecnològic, globalitzat i competitiu, amb un ritme de vida frenètic, és complicat fer que les persones dediquin al seu temps a procediments d'avaluació psicològica que impliquen certa dedicació i poden resultar pesats i avorrits. És per això que cal adaptar aquestes eines a l'era actual, permeten que els pacients en facin ús des de l'aparell que més utilitzen, el seu dispositiu mòbil, i en el moment que més els convingui, a més d'afegir elements lúdics que amenitzin el procés i motivin el pacient a acabar.

El present treball tracta la digitalització i gamificació del qüestionari psicològic IPSAQ (*Internal, Personal and Situational Attributions Questionnaire*), amb l'objectiu principal d'amenitzar el procés de respondre qüestionaris psicològics a pacients i facilitar el seguiment i anàlisi de les respostes als psicòlegs.

Es detalla el procés de disseny, implementació, avaluació i validació preliminar de l'aplicació, tant per la banda del servidor com per la del client mòbil.

Índex

1	GLOSSARI	10
2	INTRODUCCIÓ	12
2.1	Motivació	12
2.2	Objectius	12
2.3	Abast	13
3	ANTECEDENTS	14
3.1	IPSAQ	14
3.2	Visual Studio	15
3.3	Tecnologia Flask	15
3.4	Bootstrap	16
3.5	Android Studio	17
3.6	MySQL	17
3.7	SQLite	17
4	DISSENY	18
4.1	Arquitectura	18
4.2	Requeriments	18
4.2.1	Requeriments funcionals generals del servidor	19
4.2.2	Requeriments funcionals generals de l'app mòbil	19
4.3	Casos d'ús	19
4.3.1	Casos d'ús del servidor	19
4.3.2	Casos d'ús del client	27
4.4	Model de dades	31
4.4.1	Model de dades del servidor	31
4.4.2	Model de dades del client	33
4.5	Diagrames de funcionament	35
4.5.1	Diagrama de la web	36
4.5.2	Flux de l'app mòbil	37
4.6	Gamificació	38
4.6.1	Anàlisi de les alternatives	38
4.6.2	Mecanisme de gamificació escollit	39
4.7	Implementació	41
4.7.1	Servidor	41
4.7.1.1	Bootstrap	41
4.7.1.2	Estructura de blueprints	42
4.7.1.3	Flask-WTF	42

4.7.1.4	Flask–Bootstrap	43
4.7.1.5	Flask–SQLAlchemy	43
4.7.1.6	Flask–Security	43
4.7.1.7	Flask–Mail	44
4.7.1.8	Flask–Debugtoolbar	46
4.7.1.9	Flask–BabelEx	46
4.7.1.10	Flask–Excel	47
4.7.1.11	PyJWT	47
4.7.1.12	Connexions amb l'API REST	47
4.7.2	Client	47
4.7.2.1	Classes de gestió	47
4.7.2.2	Activitats	49
4.7.2.3	Extres	50
4.7.2.4	Joc	51
4.7.2.5	Menú desplegable	52
4.7.2.6	Scrolling vertical	53
4.7.2.7	Suport multilingüe	53
4.7.2.8	Connexions amb l'API REST	53
4.8	Versions de l'aplicació mòbil	54
4.8.1	Versió gamificada	54
4.8.2	Versió no gamificada	55
5	DESPLEGAMENT	57
6	PROVES	61
6.1	Proves unitàries del servidor	61
6.2	Proves unitàries de l'aplicació mòbil	61
6.3	Proves funcionals	61
7	VALIDACIÓ PRELIMINAR	62
7.1	Preguntes de l'enquesta	62
7.2	Anàlisi dels resultats	64
8	PLANIFICACIÓ I ANÀLISI ECONÒMICA	68
8.1	Planificació i realització	68
8.2	Pressupost	70
8.2.1	Recursos humans	70

8.2.1.1 Hardware.....	72
8.2.1.2 Software.....	73
8.2.1.3 Costos indirectes.....	73
8.2.2 Llicències	74
8.2.3 Total.....	76
9 IMPACTE AMBIENTAL I SOCIAL	77
CONCLUSIONS	78
AGRAÏMENTS	79
BIBLIOGRAFIA	80
Referències bibliogràfiques	80
Bibliografia complementària	80

Índex de figures

Figura 1. Arquitectura de l'aplicació.....	18
Figura 2. Casos d'ús del servidor	20
Figura 3. Casos d'ús de l'usuari	27
Figura 4. Diagrama entitat-relació de la base de dades del servidor (mySQL).....	31
Figura 5. Diagrama entitat-relació de la base de dades del client (SQLite)	33
Figura 6. Pantalles del joc entre preguntes	40
Figura 7. Pantalla final del joc.....	40
Figura 8. Fitxer HTML del formulari d'editar usuari.....	43
Figura 9. Fitxer HTML del formulari d'afegir usuari.....	43
Figura 10. Mòdul <i>emails</i>	45
Figura 11. Mòdul <i>decorators</i>	45
Figura 12. Part en castellà del correu de benvinguda.....	45
Figura 13. Funció “get_locale” de Flask-babelEx	46
Figura 14. Integració del <i>layout</i> en l' <i>AlertDialog</i>	48
Figura 15. Part del mòdul que comprova si la resposta és correcta	52
Figura 16. Menú desplegable en la pantalla principal.....	53
Figura 17. Establiment del temps d'espera i enviament de la petició a través de <i>Volley</i> .	54
Figura 18. Versió sense gamificar de l'aplicació	55
Figura 19. Configuració base de dades MySQL.....	57
Figura 20. Desplegament dels fitxers de l'app web al servidor	58
Figura 21. Creació i configuració de l'entorn virtual	58
Figura 22. Addició de les traduccions al català i castellà de Flask-Security.....	59
Figura 23. Configuració del servidor	59

Figura 25. Percentatge d'ús de cada versió	64
Figura 26. Aplicació gamificada vs. no gamificada	65
Figura 27. Gràfiques de les preguntes relatives al joc	66
Figura 28. Comentaris i suggeriments rebuts	66
Figura 29. Diagrames de Gant de la planificació (a dalt) i realització (a baix) del treball .	69

1 Glossari

- IPSAQ: *Internal Personal and Situational Attributions Questionnaire* (qüestionari d'atribucions internes, personals i situacionals).
- ASQ: *Attributional Style Questionnaire* (qüestionari d'estil atribucional).
- EB: *Externalizing Bias* (biaix externalitzant).
- PB: *Personalizing Bias* (biaix personalitzant).
- HTML: *HyperText Markup Language* (llenguatge de marques d'hipertext).
- ORM: *Object Relational Mapper* (mapejador relacional d'objectes).
- SMTP: *Simple Mail Transfer Protocol* (protocol per a la transferència simple de correus).
- IDE: *Integrated Development Environment* (entorn de desenvolupament integrat).
- DB: Database (base de dades).
- API: *Application Programming Interface* (Interfície de programació d'aplicacions).
- REST: *Representational state transfer* (Transferència d'estat representacional).
- Id: identificador.
- Framework: entorn o marc de treball.
- TXT: *textfile* (arxiu de text).
- CSS: *Cascading Style Sheets* (Fulles d'estil en cascada).
- JS: JavaScript.
- XML: *Extensible Markup Language* (Llenguatge de marcat extensible).
- Activitat: component de l'aplicació que conté una única pantalla amb una interfície d'usuari amb la qual es pot interactuar per realitzar una acció.
- Intent: objecte d'acció utilitzat per sol·licitar una acció d'un altre component de l'aplicació.
- Extra: és un Bundle (mapejador de claus en format string a valors) i s'utilitza per

proporcionar qualsevol informació extra al component associat a un intent.

- HTTP: Hypertext Transfer Protocol (protocol de transferència d'hipertext).
- HTTPS: Hypertext Transfer Protocol Secure (protocol segur de transferència d'hipertext).

2 Introducció

2.1 Motivació

Actualment, l'avaluació psicològica i psiquiàtrica dels pacients acostuma a realitzar-se a través de tests estàndards. Un d'aquests tests és l'*Internal, Personal and Situational Attributions Questionnaire* (IPSAQ), que pretén mesurar el perfil atributiu de les persones, és a dir, la tendència de la persona a atribuir l'origen d'una situació a un mateix, als altres o a causes exteriors.

Amb la creixent globalització i l'auge de les noves tecnologies, sorgeix la necessitat d'adaptar aquesta eina a l'era actual, mitjançant la digitalització del procés. Això permet la recollida automàtica de les respostes i facilita la seva gestió al llarg del temps.

Sovint els tests utilitzats són pesats i avorrits de fer. Això planteja la possibilitat d'introduir mecanismes de gamificació, és a dir, mecàniques de joc en el qüestionari per tal d'amenitzar-lo.

El treball de fi de grau aquí presentat s'emmarca en una col·laboració entre el CREB i el Parc Sanitari Sant Joan de Déu, que neix amb el propòsit d'explorar la possibilitat de crear qüestionaris d'avaluació psicològica digitals més amens.

2.2 Objectius

L'objectiu d'aquest projecte és el disseny, implementació i validació preliminar d'una aplicació gamificada que permeti passar el qüestionari psicològic d'estil atribucional IPSAQ a pacients i avaluar-ne els resultats. L'objectiu es desglossa en els següents:

- Anàlisi dels requisits.
- Disseny i creació d'un servidor on emmagatzemar les dades dels usuaris i les seves respostes.
- Disseny i implementació d'una app web multilingüe estèticament bonica on el psicòleg pugui gestionar la creació d'usuaris i administradors i analitzar respostes.
- Disseny i implementació d'una app mòbil multilingüe que permeti respondre el qüestionari IPSAQ.
- Dissenyar dues versions de l'app mòbil, una amb gamificació i l'altra sense, per tal de poder analitzar si la introducció d'elements de joc millora l'experiència d'usuari.
- Estudi preliminar d'avaluació.

- Validació preliminar. Es realitzarà una primera fase de tests amb usuaris escollits. La validació amb pacients queda fora de l'abast del TFG.

2.3 Abast

L'aplicació està dissenyada per funcionar només en el sistema operatiu Android. S'ha escollit aquest sistema operatiu per diversos motius:

- És un sistema obert i lliure, és a dir, gairebé no imposa restriccions quant a la programació d'una aplicació per poder ser publicada, la qual cosa facilita molt el disseny i programació en aquest sistema.
- És molt més barat que els seus competidors principals. Mentre que la llicència per ser desenvolupador en Android és de 25\$ en un únic pagament, la de iOS costa 99\$ anuals.
- És el sistema operatiu que domina el mercat mundial actualment. Segons una estadística de Statista [1], el 88% de tots els telèfons mòbils venuts mundialment en el segon quadrimestre del 2018 eren de sistema operatiu Android.

En tractar-se d'una prova de concepte, l'app no es publicarà a Google Store.

3 Antecedents

3.1 IPSAQ

L'IPSAQ és un qüestionari psicològic d'estil atribucional que mesura de manera explícita el locus de control dels pacients, és a dir, la seva percepció sobre quin és l'agent causant dels esdeveniments de la seva vida quotidiana.

L'IPSAQ apareix per primer cop en un article publicat al 1996 [2] per Peter Kinderman i Richard P. Bentall en el Departament de psicologia clínica de la universitat de Liverpool. En ell, els autors presenten una nova manera de mesurar el locus de control, que pretén confirmar la seva relació amb desordres psicològics severos, particularment la paranoia. Exposen, a més, la seva intenció de solucionar el problema de baixa fiabilitat dels anteriors qüestionaris d'estil atribucional, en especial l'*Attributional Style Questionnaire* (ASQ), i d'aprofitar la potencial utilitat de les taxonomies no incloses en aquests.

El qüestionari distingeix 3 tipus de locus o atribucions: interna (relacionada amb el propi subjecte), externa personal (relacionada amb altres persones) o externa situacional (relacionada amb les circumstàncies o l'atzar).

El qüestionari consta de 32 preguntes (veure annex 1) que descriuen 16 situacions socials positives i 16 negatives. El pacient ha d'identificar quina és la causa més probable per a cada situació descrita i categoritzar-la dins d'un dels 3 grups atribucionals especificats.

A partir de les respostes, l'IPSAQ permet generar 6 subescales amb un grau de fiabilitat acceptable i extreure 2 biaixos cognitius, l'EB o biaix externalitzant i el PB o biaix personalitzant. L'EB s'obté sostraint el nombre d'atribucions internes per a esdeveniments negatius del nombre d'atribucions internes per a esdeveniments positius. El PB representa la proporció d'atribucions externes que són personals en comptes de situacionals. Aquestes 2 mesures permeten categoritzar el pacient dins d'un dels 3 grups següents:

- Atribuïdor intern: $EB \leq 0$.
- Atribuïdor personal: $EB > 0$, $PB > 0,5$.
- Atribuïdor situacional: $EB > 0$, $PB \leq 0,5$.

Els creadors del qüestionari van determinar que les atribucions internes per a esdeveniments negatius podien estar relacionades amb mal humor i depressió, les atribucions personals amb ideacions paranoiques i les atribucions situacionals semblaven ser psicològicament benignes.

3.2 Visual Studio

Pel desenvolupament de l'aplicació web s'ha fet servir el software gratuït Visual Studio Community 2019, un *Integrated Development Environment* (IDE) força potent que incorpora diverses de les eines utilitzades en el projecte (Flask, Flask-Bootstrap...) i dreceres que simplifiquen i fan més ràpida l'edició de codi.

3.3 Tecnologia Flask

Flask és un microframework *Open Source* desenvolupat en Python que permet desenvolupar aplicacions web amb tecnologia MVC. Es categoritza com a microframework perquè el seu nucli és molt senzill i només inclou les eines mínimes i bàsiques per al desenvolupament d'una app web funcional, però compta amb un gran potencial d'ampliació, ja que disposa d'una gran varietat d'extensions i complements que permeten afegir noves funcionalitats. Així, només cal instal·lar les funcionalitats que realment són necessàries en l'aplicació.

Gràcies a les seves característiques, Flask és més lleuger, ràpid i pràctic que altres de les possibilitats que hi ha al mercat, com Django, el framework web més famós en l'actualitat. Tot això permet obtenir una app menys pesada i més veloç, amb un codi més simple i fàcil d'entendre, la qual cosa facilita el seu manteniment. A més, Flask compta amb una documentació [3] molt amplia. Per tot això, s'ha escollit aquest entorn de treball per al desenvolupament de l'app web.

Flask integra el motor de plantilles Jinja2, que permet renderitzar plantilles i generar documents HTML vàlids. A més de permetre introduir instruccions i variables en els fitxers HTML permet utilitzar plantilles que poden ser heretades i utilitzades per cada vista en particular. Això s'aconsegueix amb l'ús de blocs reemplaçables.

A més del nucli principal, s'han utilitzat les extensions següents:

- **Flask-WTF:** integra l'aplicació Flask amb WTForms¹. Permet crear i validar formularis, essent necessària una classe per a cadascun d'ells.
- **Bootstrap-Flask:** Proporciona plantilles base pels fitxers HTML creats. Permet renderitzar camps aïllats o formularis complets creats amb Flask-WTF o WTForms.

¹ WTForms és una llibreria de Python que permet renderitzar i validar formularis.

- **Flask-SQLAlchemy:** afegeix suport per l'*Object Relational Mapper* (ORM) SQLAlchemy dins l'aplicació. Permet integrar l'aplicació web amb una base de dades mitjançant la creació de models que relacionen les taules de la base de dades amb classes de Python.
- **Flask-Security:** integra diverses extensions de Flask i permet afegir de manera ràpida diferents mecanismes de seguretat. Proporciona un magatzem de dades (*datastore*) d'usuaris integrat amb SQLAlchemy que facilita la creació i la consulta d'usuaris i les seves dades. A més, simplifica la gestió dels rols, proporciona funcions per encriptar contrasenyes, proveeix autenticació basada en sessió o en token, autenticació HTTP bàsica, etc.
- **Flask-Mail:** proveeix una interfície que permet configurar un *Simple Mail Transfer Protocol* (SMTP) per tal de permetre enviar correus electrònics des de dins de l'aplicació.
- **Flask-Debugtoolbar:** Afegeix una barra d'eines a les respostes HTML de l'aplicació si es té el mode de desenvolupament o debug activat. S'utilitza a l'hora de programar l'aplicació per tal de verificar-ne l'estat i arreglar errors.
- **Flask-BabelEx:** proporciona suport multilingüe a l'aplicació. Permet extreure i traduir manualment a qualsevol idioma els texts marcats per tal de poder mostrar l'aplicació en la llengua corresponent.
- **Flask-Excel:** permet consumir o produir dades guardades en fitxers d'Excel dins l'aplicació, ja sigui a través de taules senceres, columnes d'una base de dades o estructures de Python.

3.4 Bootstrap

Bootstrap és un framework de codi obert que proporciona plantilles base pels fitxer HTML d'una app web i diverses eines de disseny, com formularis, botons o menús desplegable, basades en HTML i CSS, així com extensions de JavaScript addicionals. Bootstrap permet crear un disseny web adaptatiu ("responsive"), és a dir, que s'ajusta automàticament a les característiques del dispositiu utilitzat, ja siguin telèfons mòbils, ordinadors o tabletas, podent-se adaptar a la seva mida i resolució. A més, dóna lloc a dissenys simples i intuïtius i és compatible amb tots els navegadors moderns (Google Chrome, Mozilla Firefox, Internet Explorer, Safari i Opera). Tot això, sumat a la seva extensa documentació [4], el converteix en la opció idònia per desenvolupar la part gràfica de l'app web.

3.5 Android Studio

Pel desenvolupament de l'app mòbil s'ha fet ús de l'IDE oficial de la plataforma Android: Android Studio. A més de ser un software gratuït i disposar d'una interfície específica pel desenvolupament en Android, conté moltes funcions que ajuden a optimitzar la programació: detecció en temps real d'errors sintàctics, de compatibilitat o rendiment, consells d'optimització, renderitzat en temps real, possibilitat d'executar l'app al mòbil, refactorització, etc.

Quant als llenguatges de programació, s'han empleat els 2 més utilitzats per a programació en Android: Java per a la part funcional i XML per al disseny gràfic.

3.6 MySQL

MySQL és un sistema de gestió de bases de dades relacional. És de codi obert i dels més populars actualment, en especial per a entorns de desenvolupament web.

3.7 SQLite

SQLite és un sistema de gestió de bases de dades relacional. Té la particularitat que s'integra amb el programa en què s'utilitza, de manera que s'accedeix a les seves funcionalitats a través de simples crides a funcions. Aquestes són més eficients que la comunicació entre processos, que es dona en el cas de les connexions a bases de dades clients-servidor, de manera que es redueix així el temps de resposta. S'ha decidit implementar aquest sistema en l'aplicació mòbil per tal de millorar la navegabilitat a través seu i reduir els temps d'espera.

4 Disseny

4.1 Arquitectura

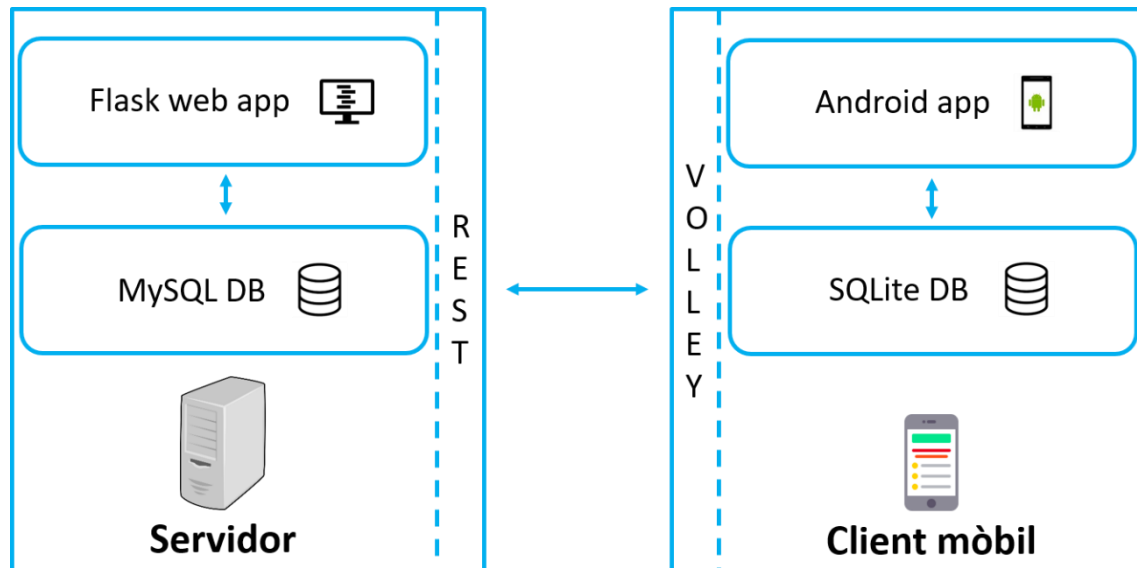


Figura 1. Arquitectura de l'aplicació

Com mostra l'esquema anterior, l'arquitectura de l'aplicació consta fonamentalment d'un client mòbil i un servidor que es connecten entre sí. El servidor està format per una app web creada amb Flask, on l'administrador pot fer les seves gestions i consultar dades, i una base de dades MySQL on s'emmagatzema la informació. El client mòbil està constituït per una aplicació Android i una base de dades SQLite, on s'emmagatzemen dades relatives a l'aplicació o altres dades de manera temporal fins que són enviades al servidor. D'aquesta manera, l'aplicació no ha de connectar-se constantment al servidor i el funcionament es fa més fluid. Totes les connexions entre servidor i client es fan mitjançant tecnologia REST. Particularment, des de la banda del client s'utilitza la llibreria *Volley* d'Android, que implementa aquesta tecnologia. Des de la banda del servidor, s'utilitza la funció *jsonify* de flask per tal de convertir diccionaris en objectes JSON i enviar-los al client, i la funció *get_json* del mòdul *request* de flask per tal d'obtenir l'objecte JSON provinent el client.

4.2 Requeriments

A continuació es llisten els principals requeriments funcionals del sistema. El detall d'aquests requeriments està desglossat en els casos d'ús de l'apartat o. Les funcionalitats es divideixen en 2 grups: les del servidor i les de l'app mòbil.

4.2.1 Requeriments funcionals generals del servidor

- Disposar d'una base de dades on emmagatzemar la informació.
- Disposar d'un lloc web per la gestió d'usuaris i qüestionaris amb les següents funcionalitats principals:
 - o Permetre la gestió (creació, edició i eliminació) d'usuaris.
 - o Permetre la consulta, descàrrega i eliminació dels resultats dels qüestionaris realitzats.
 - o Permetre la consulta i descàrrega de les respostes de cada qüestionari.
 - o Control de permisos d'usuari.
 - o Suport multilingüe: la web ha d'estar disponible, com a mínim, en català i castellà i, si és possible, també en anglès.

4.2.2 Requeriments funcionals generals de l'app mòbil

- Disposar d'una base de dades on emmagatzemar informació temporalment.
- Disposar d'una app mòbil en què els usuaris puguin respondre el qüestionari IPSAQ. L'app ha de disposar, com a mínim, de les següents funcionalitats:
 - o Incorporar mecanismes de gamificació per tal d'amenitzar el qüestionari i incentivar els usuaris a acabar-lo.
 - o Enviar els resultats al servidor.
 - o Mostrar als usuaris el resultat del qüestionari en finalitzar.
 - o Control de permisos d'usuari.
 - o Sistema de gestió de l'oblit de contrasenya per part dels usuaris.
 - o Suport multilingüe: l'app ha d'estar disponible, com a mínim, en català i castellà i, si és possible, també en anglès.

4.3 Casos d'ús

4.3.1 Casos d'ús del servidor

Hi ha 3 rols definits dins del servidor:

- **Administrador:** Es connecta a la web i pot realitzar diverses funcions d'administració.
- **Superadministrador:** Pot realitzar totes les funcions de la web, incloent l'administració d'administradors.
- **Usuari:** es connecta només des de l'app mòbil, de forma indirecta i transparent

mitjançant una API REST.

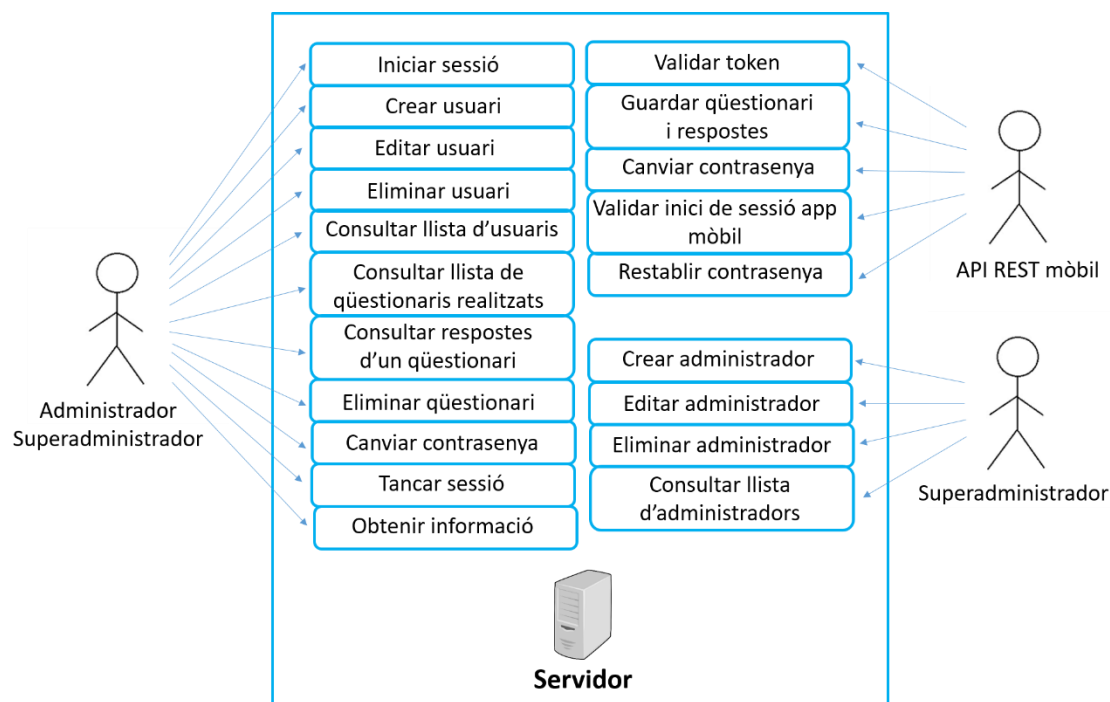


Figura 2. Casos d'ús del servidor

En el diagrama anterior, es poden veure els diferents casos d'ús del servidor per part dels diferents subjectes que poden connectar-s'hi. Aquests són: l'administrador i el superadministrador a través de la web app i la part de l'API REST que envia peticions des del mòbil. Com queda reflectit en la figura, el superadministrador pot realitzar totes les funcions de l'administrador i altres. Es descriuen els diferents casos d'ús del servidor a continuació.

Cas d'ús	Iniciar sessió
Descripció	Cal autenticar-se dins la web per tal d'accedir a les funcions d'administrador i superadministrador
Subjectes	Administrador i superadministrador.
Condicions d'entrada	-

Es comprova que l'usuari estigui registrat dins la base de dades amb rol d'administrador o superadministrador i que la contrasenya introduïda sigui correcta.

Cas d'ús	Crear usuari
Descripció	Registrar un usuari amb el seu email i una contrasenya perquè pugui iniciar sessió a l'app mòbil.
Subjectes	Administrador i superadministrador.
Condicions d'entrada	Haver iniciat sessió com a administrador o superadministrador.

S'envia un correu de benvinguda a l'usuari creat amb la contrasenya escollida des de l'adreça ipsappcontact@gmail.com.

Cas d'ús	Editar usuari
Descripció	Canviar la contrasenya d'un usuari o la seva activació.
Subjectes	Administrador i superadministrador.
Condicions d'entrada	Haver iniciat sessió com a administrador o superadministrador.

Si es canvia la contrasenya, s'envia un correu a l'usuari editat amb la nova contrasenya des de l'adreça ipsappcontact@gmail.com.

Cas d'ús	Eliminar usuari
Descripció	Eliminar un usuari de l'aplicació.
Subjectes	Administrador i superadministrador.
Condicions d'entrada	Haver iniciat sessió com a administrador o superadministrador. Cal haver eliminat abans qualsevol qüestionari que tinguis associat.

No es podrà eliminar un usuari que tingui qüestionaris.

Cas d'ús	Consultar llista d'usuaris
Descripció	Veure taula amb els emails de tots els usuaris registrats, la seva activació i el seu rol.
Subjectes	Administrador i superadministrador.
Condicions d'entrada	Haver iniciat sessió com a administrador o superadministrador.

Cas d'ús	Consultar llista de qüestionaris realitzats
Descripció	Veure taula amb el correu electrònic, data, EB i PB de tots els qüestionaris enviats des de l'aplicació mòbil.
Subjectes	Administrador i superadministrador.
Condicions d'entrada	Haver iniciat sessió com a administrador o superadministrador.

Cas d'ús	Consultar respostes d'un qüestionari
Descripció	Veure l'opció escollida, la causa i el percentatge introduïts per a totes les preguntes del test seleccionat.
Subjectes	Administrador i superadministrador.
Condicions d'entrada	Haver iniciat sessió com a administrador o superadministrador.

Cas d'ús	Eliminar qüestionari
Descripció	Eliminar un qüestionari del servidor.
Subjectes	Administrador i superadministrador.
Condicions d'entrada	Haver iniciat sessió com a administrador o superadministrador.

S'eliminen automàticament totes les preguntes associades al qüestionari seleccionat.

Cas d'ús	Canviar contrasenya
Descripció	Els administradors i superadministradors poden canviar la seva contrasenya des de l'app web
Subjectes	Administrador i superadministrador.
Condicions d'entrada	Haver iniciat sessió com a administrador o superadministrador.

Cal introduir la mateixa contrasenya 2 cops perquè la petició de canvi de contrasenya es processi amb èxit.

Cas d'ús	Tancar sessió
Descripció	Deixar sense efecte el login de l'usuari
Subjectes	Administrador i superadministrador.
Condicions d'entrada	Haver iniciat sessió com a administrador o superadministrador.

Cas d'ús	Obtenir informació
Descripció	Accedir a pantalles exclusivament informatives, com la d'inici i presentació de la web.
Subjectes	Tothom
Condicions d'entrada	-

Cas d'ús	Validar token
Descripció	Comprovar que el token de l'usuari rebut és vàlid en el servidor.
Subjectes	API REST
Condicions d'entrada	-

En cas que el token es trobi dins del *header* de la petició rebuda, es descripta amb la clau secreta de l'aplicació i s'obté l'usuari que hi correspon. Si no és vàlid o està caducat s'obté un error.

Cas d'ús	Guardar qüestionari i respostes
Descripció	Guardar l'email de l'usuari, la data i hora d'enviament i les respostes de les preguntes del qüestionari en el servidor.
Subjectes	API REST.
Condicions d'entrada	Token de l'usuari vàlid en el servidor.

Cas d'ús	Canviar contrasenya
Descripció	Canviar la contrasenya d'un usuari dins la base de dades del servidor.
Subjectes	API REST.
Condicions d'entrada	Token de l'usuari vàlid en el servidor.

Cas d'ús	Validar inici de sessió app mòbil
Descripció	Comprovar que l'email introduït està registrat i que la contrasenya és correcta.
Subjectes	API REST
Condicions d'entrada	-

Si el correu electrònic rebut pel servidor es troba registrat a la base de dades i la contrasenya enviada és la que hi correspon, es genera un nou token que caduca al cap de 7 dies.

Cas d'ús	Restablir contrasenya
Descripció	Canviar la contrasenya d'un usuari que l'ha oblidat.
Subjectes	API REST
Condicions d'entrada	-

En primer lloc, es verifica que el correu electrònic rebut pel servidor existeix a la base de dades i, si es així, es genera una contrasenya alfanumèrica de 8 caràcters aleatòria. S'actualitza la base de dades del servidor atorgant la nova contrasenya a l'usuari amb el correu electrònic rebut a la petició i se li envia un mail des de l'adreça ipsappcontact@gmail.com.

Cas d'ús	Crear administrador
Descripció	Registrar un administrador amb el seu email i una contrasenya perquè pugui iniciar sessió a l'app web.
Subjectes	Superadministrador.
Condicions d'entrada	Haver iniciat sessió com a superadministrador.

S'envia un correu de benvinguda a l'administrador creat amb la seva contrasenya des de l'adreça ipsappcontact@gmail.com

Cas d'ús	Editar administrador
Descripció	Canviar la contrasenya d'un administrador, modificar la seva activació o el seu rol.
Subjectes	Superadministrador.
Condicions d'entrada	Haver iniciat sessió com a superadministrador.

Es comprova si s'ha canviat la contrasenya i, si és així, s'envia un correu a l'administrador editat amb la nova contrasenya des de l'adreça ipsappcontact@gmail.com.

Cas d'ús	Eliminar administrador
Descripció	Eliminar les dades d'un administrador del servidor.
Subjectes	Superadministrador.
Condicions d'entrada	Haver iniciat sessió com a superadministrador.

Cas d'ús	Consultar llista d'administradors
Descripció	Veure taula amb el correu electrònic de tots els administradors registrats, la seva activació i el seu rol.
Subjectes	Superadministrador.
Condicions d'entrada	Haver iniciat sessió com a superadministrador.

4.3.2 Casos d'ús del client

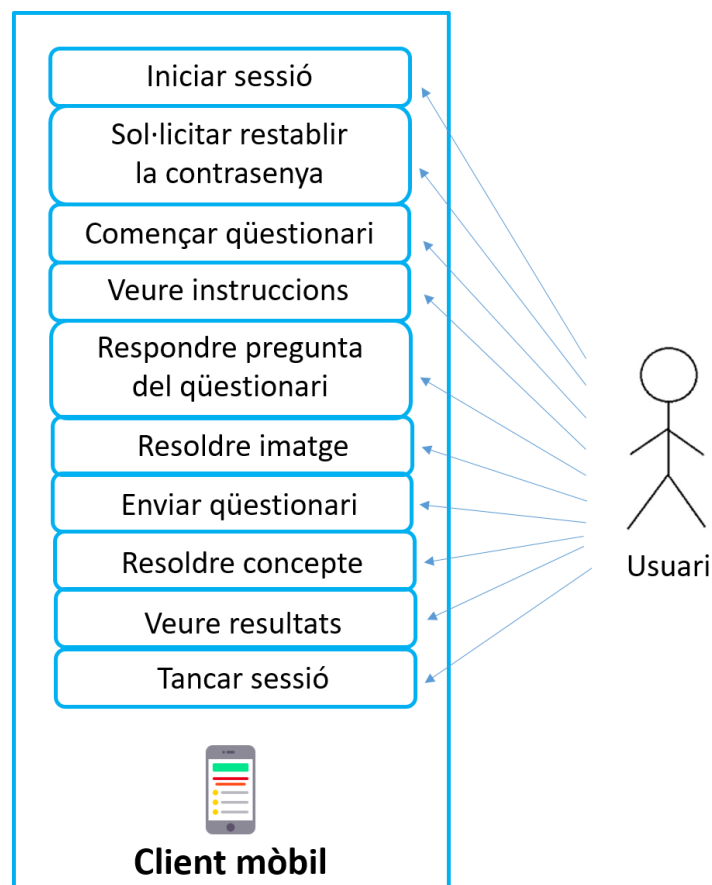


Figura 3. Casos d'ús de l'usuari

La figura 3 reflecteix els diferents casos d'ús de l'app mòbil per part de l'únic subjecte que en pot fer ús, l'usuari. Es descriuen a continuació.

Cas d'ús	Iniciar sessió
Descripció	Cal autenticar-se dins l'app mòbil per poder realitzar la resta de funcions.
Subjectes	Usuari.
Condicions d'entrada	Estar registrat a la base de dades del servidor.

Si la petició és processada amb èxit, l'app mòbil rep un token que és guardat juntament amb el correu electrònic per tal de mantenir la sessió iniciada i proporcionar seguretat. Si es tracta del primer inici de sessió o l'usuari encara no ha passat d'aquesta pantalla, es requereix que completi el seu perfil introduint nom i sexe. A més es requereix que l'usuari canviï la seva contrasenya. Això es fa per raons de seguretat després que se li hagi enviat una contrasenya per mail (en registrar-lo o si l'ha oblidat).

Cas d'ús	Sol·licitar restablir la contrasenya
Descripció	L'usuari ha d'introduir el seu correu per tal de sol·licitar que se li envii una nova contrasenya per correu.
Subjectes	Usuari.
Condicions d'entrada	Estar registrat a la base de dades del servidor.

Si l'usuari està registrat i la petició és superada amb èxit, el proper cop que iniciï sessió dins l'aplicació amb la nova contrasenya haurà de canviar-la obligatòriament abans de continuar.

Cas d'ús	Començar qüestionari
Descripció	Començar el procés de respondre preguntes i resoldre imatges.
Subjectes	Usuari.
Condicions d'entrada	Haver iniciat sessió com a usuari.

Quan es comença un qüestionari, l'aplicació selecciona aleatòriament un concepte i una imatge i les guarda com a concepte i imatge actual de l'usuari. A més, es va a la pantalla

d'instruccions, a diferència de quan es reprèn un qüestionari ja començat, que s'accedeix a la darrera pregunta resposta o a la fase del joc corresponent.

Cas d'ús	Veure instruccions
Descripció	L'usuari pot veure les instruccions en qualsevol moment del qüestionari o des de la pantalla principal a través del menú desplegable situat en la barra superior de l'aplicació
Subjectes	Usuari.
Condicions d'entrada	Haver iniciat sessió com a usuari.

Cas d'ús	Respondre pregunta del qüestionari
Descripció	Es guarda l'opció escollida, la causa i el percentatge introduïts per a la pregunta corresponent, de manera permanent, en el client mòbil.
Subjectes	Usuari.
Condicions d'entrada	Haver iniciat sessió com a usuari.

Cas d'ús	Resoldre imatge
Descripció	Introduir la paraula que l'usuari creu que correspon al que s'amaga darrere de la part visible de la imatge.
Subjectes	Usuari.
Condicions d'entrada	Haver iniciat sessió com a usuari.

En cas que l'usuari hagi encertat la paraula o si ja ha desbloquejat totes les peces de la imatge, passa a veure la imatge completa i, en cas contrari, rep un missatge d'alerta que l'averteix que ha fallat abans de passar a la pregunta següent.

Cas d'ús	Enviar qüestionari
Descripció	Quan l'usuari ha finalitzat totes les preguntes, se l'adverteix i es requereix que envii el qüestionari al servidor.
Subjectes	Usuari.
Condicions d'entrada	Haver iniciat sessió com a usuari.

Si la petició és superada amb èxit, les respostes s'esborren del client mòbil, es canvia l'estat de l'usuari, i s'obtenen en el client l'EB i el PB enviats des del servidor.

Cas d'ús	Resoldre concepte
Descripció	Després d'enviar el qüestionari, l'usuari pot veure totes les imatges que ha encertat o arribat a completar i ha d'introduir el concepte que creu que les relaciona.
Subjectes	Usuari.
Condicions d'entrada	Haver iniciat sessió com a usuari.

Quan l'usuari clica "Enviar" li apareix una alerta que l'informa de si ha encertat o no, li indica la resposta correcta i li permet accedir als resultats.

Cas d'ús	Veure resultats
Descripció	Abans de finalitzar, l'usuari ha de veure els resultats del seu qüestionari.
Subjectes	Usuari.
Condicions d'entrada	Haver iniciat sessió com a usuari.

En funció del PB i l'EB obtinguts, es calcula a quin grup d'atribuïdors pertany l'usuari i se li mostra un missatge descriptiu.

Cas d'ús	Tancar sessió
Descripció	L'usuari pot tancar sessió en qualsevol moment des del menú desplegable de la barra superior de l'aplicació.
Subjectes	Usuari.
Condicions d'entrada	Haver iniciat sessió com a usuari.

S'esborren les dades de sessió.

4.4 Model de dades

4.4.1 Model de dades del servidor

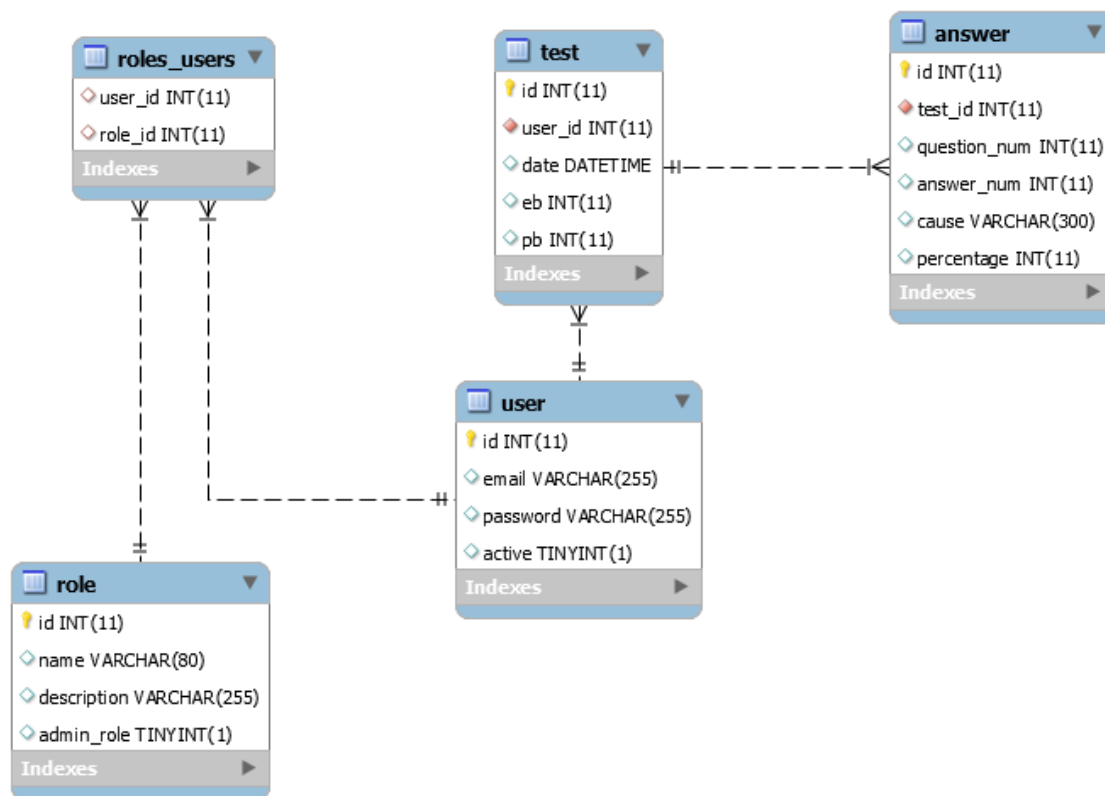


Figura 4. Diagrama entitat-relació de la base de dades del servidor (MySQL)

El diagrama *Entity-relationship* (ER) anterior s'ha fet amb MySQL Workbench i mostra el model de base de dades del servidor. En l'annex 2 pot consultar-se la interpretació dels símbols del diagrama (línies i icones de cada columna). Poden observar-se 5 taules diferents:

Taula	Descripció
user	Conté les dades dels usuaris de tota l'aplicació: identificador, correu electrònic, contrasenya, i un paràmetre que indica si estan actius (valor 0 si no ho estan i 1 si ho estan).
role	Inclou els diferents rols, que venen definits per un identificador, el seu nom, una descripció i un paràmetre representatiu d'un booleà que indica si es tracta d'un rol d'administrador (1) o no (0).
roles_users	Relaciona els usuaris de tota l'aplicació amb els seus rols, a través de l'identificador de l'usuari i el del rol que té associat.
test	Conté tots els qüestionaris realitzats, definits pel seu identificador, el de l'usuari que l'ha realitzat, la data i hora d'enviament i l'EB i el PB associats.
answer	Inclou totes les respostes dels qüestionaris, amb el seu identificador, l'identificador del test al qual pertany, el número de pregunta, el número de l'opció escollida (1 per a l'opció a, 2 per a la b i 3 per a la c) i la causa i percentatge introduïts.

Totes les taules menys la de *roles_users* s'han creat com a classes de Python que hereten de la classe *Model* de SQLAlchemy. A més, les classes *User* i *Role* hereten de les classes *UserMixin* i *RoleMixin* de Flask-Security respectivament, la qual cosa facilita la creació i gestió d'usuaris i rols, a través de la implementació d'un magatzem de dades (*datastore*) integrat amb SQLAlchemy. També simplifica la creació de rutes, ja que permet l'ús de decoradors com *roles_accepted* o *login_required*, la qual cosa limita l'accés a certes rutes a usuaris amb determinats rols o amb sessió iniciada.

La taula *roles_users* s'utilitza com a taula intermèdia per tal de permetre una relació n:n entre usuaris i rols sense haver de tenir les dades dels usuaris duplicades.

En la pràctica, només hi ha 3 rols definits: usuari (de l'app mòbil), administrador i superadministrador. Així, la relació 1:n entre rols i rols d'usuaris serà sempre 1:3 i la relació entre usuaris i rols d'usuaris estarà limitada a un màxim de 1:3. En la base de dades s'han representat ambdues relacions com a 1:n a causa de la manca d'aquestes limitacions a l'hora de definir el model. De la mateixa manera, la relació 1:n entre tests i respostes serà sempre d'1 a 32.

4.4.2 Model de dades del client

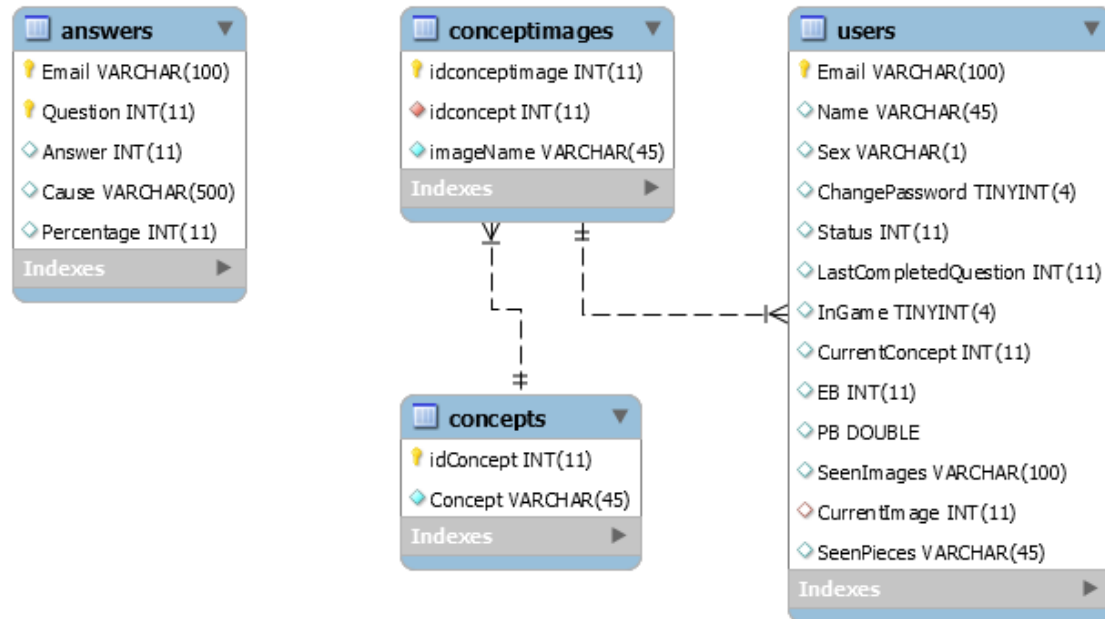


Figura 5. Diagrama entitat-relació de la base de dades del client (SQLite)

El diagrama entitat-relació anterior s'ha fet amb MySQL Workbench i mostra 4 taules en la base de dades del client:

Taula	Descripció
users	Taula d'usuaris de l'app mòbil. Conté l'email de l'usuari com a identificador, el seu nom, sexe, el número de l'última pregunta que ha respòs (0 per defecte), un paràmetre que representa un booleà i indica si l'usuari ha de canviar la seva contrasenya (0 -per defecte- si no l'ha de canviar i 1 altrament), un enter que indica l'estat de l'aplicació, un paràmetre representatiu d'un booleà que indica si l'usuari es trobava dins d'una pregunta (0) o del joc (1) en abandonar el qüestionari, l'identificador del concepte que s'ha seleccionat pel joc, el de la imatge actual del joc, l'EB i el PB del qüestionari (0 per defecte), i els paràmetres <i>seenImages</i> i <i>seenPieces</i> , que contenen, respectivament, les imatges ja mostrades a l'usuari en aquell qüestionari i l'índex de les peces de la imatge actual que ja ha vist, per tal d'evitar repeticions.
concepts	Conté els conceptes seleccionats pel joc, amb el seu identificador i el seu nom.
conceptImages	Conté les imatges associades als diferents conceptes del joc, definides pel seu identificador, el del concepte al qual estan associades i el seu nom (amb l'extensió inclosa).
answers	Conté les respostes a les diferents preguntes del qüestionari, amb el número de pregunta i el correu electrònic de l'usuari com a identificadors. A més, inclou l'opció escollida per l'usuari (1 per a l'opció a, 2 per a la b i 3 per a la c), la causa i el percentatge introduïts.

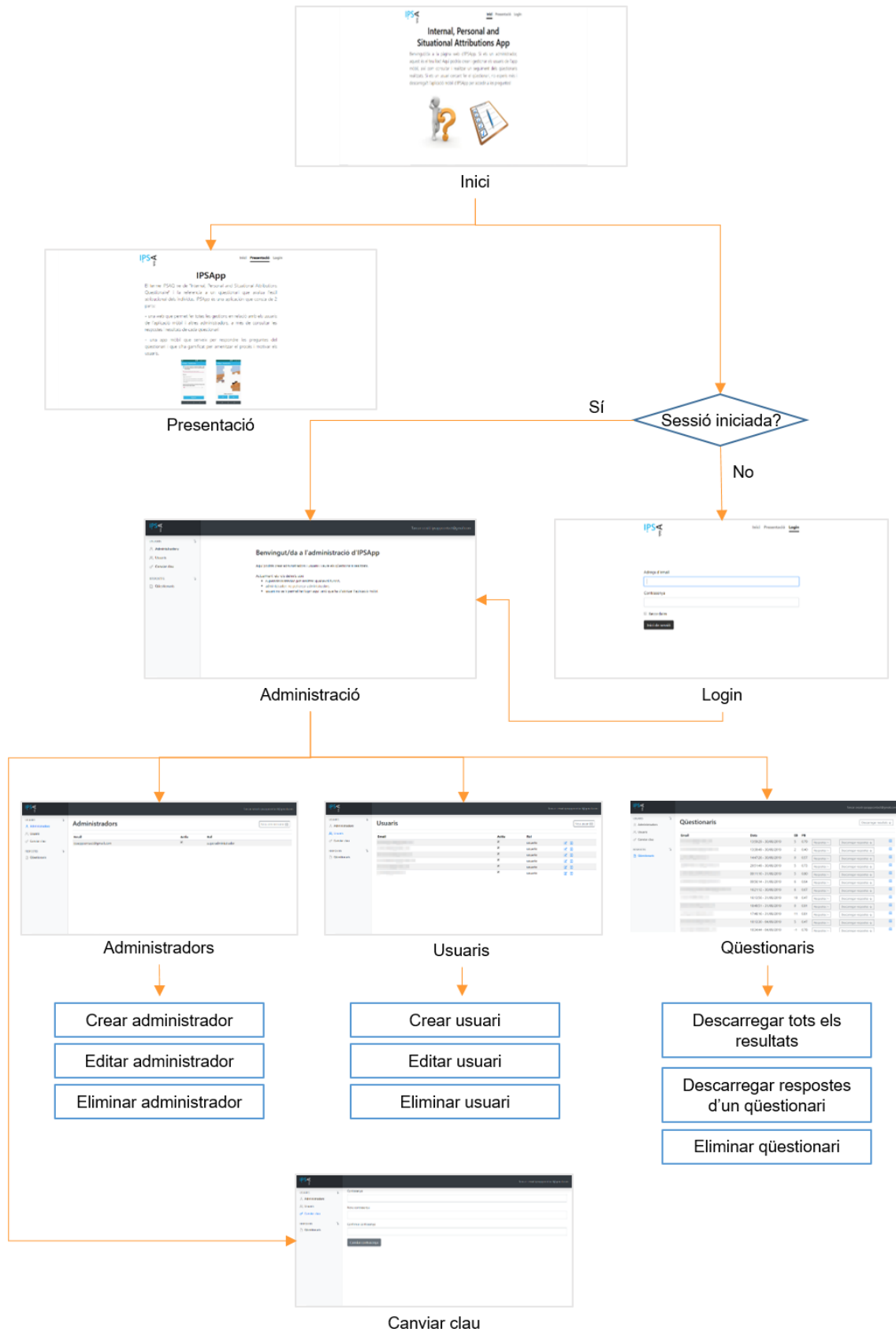
L'estat de l'aplicació guardat a la taula d'usuaris és 0 (estat inicial) si el subjecte no ha començat cap qüestionari, 1 si en té un a mitges, 2 si l'ha completat però no enviat, 3 si l'ha completat i enviat però no ha respòs la pregunta final i 4 si ho ha completat tot i pot començar-ne un de nou.

Els paràmetres *seenImages* i *seenPieces* de la taula d'usuaris són strings que contenen nombres separats per un punt i coma. En el primer cas, cada nombre entre punts i coma és l'identificador d'una imatge ja vista en aquell qüestionari, i en el segon cas, és l'índex d'una peça ja mostrada per a la imatge actual. Els paràmetres estan pensats així per tal de poder ser convertits de manera eficient i fàcil en llistes o arrays i comparar si un identificador o índex obtingut s'hi troba a dins.

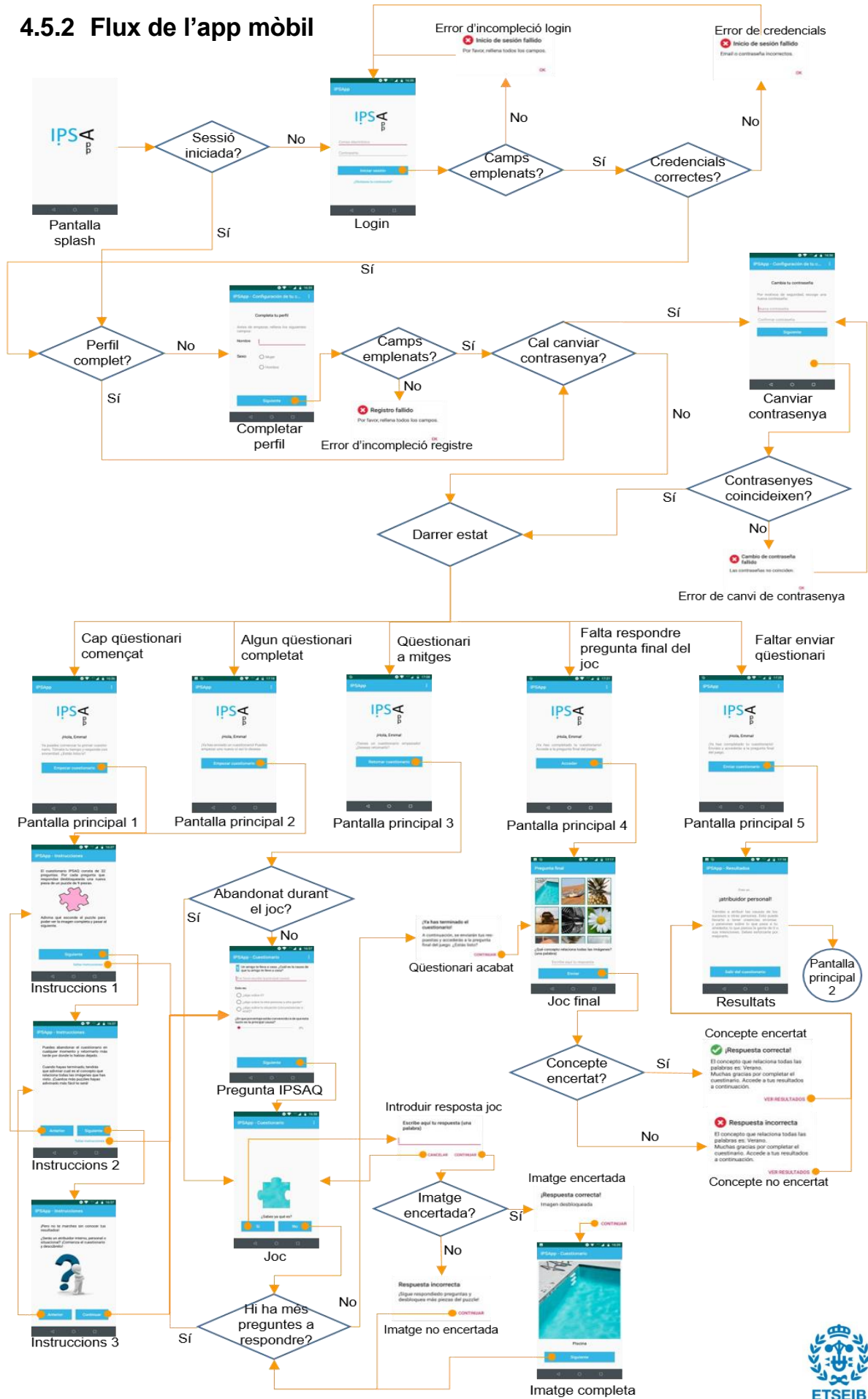
4.5 Diagrames de funcionament

A continuació es presenten els diagrames de funcionament de les dues parts de l'aplicació. L'app mòbil presenta un flux preestablert, per la qual cosa s'ha descrit amb un diagrama de flux, mentre que en el cas de la web el diagrama reflecteix les diferents opcions que té l'usuari en accedir-hi. En ambdós casos s'han utilitzat captures de pantalla reals de l'aplicació, que s'adjunten a escala major en els annexos 4 i 5.

4.5.1 Diagrama de la web



4.5.2 Flux de l'app mòbil



4.6 Gamificació

4.6.1 Anàlisi de les alternatives

El procés de respondre el qüestionari IPSAQ pot ser percebut com a una tasca avorrida pels pacients. Aquest qüestionari consta de 32 preguntes de naturalesa similar i, per tant, poden semblar repetitives. A més de l'elevat nombre de preguntes, cada pregunta implica, no només una selecció, sinó també escriure un text i valorar un aspecte sobre aquest. Per últim, algunes preguntes poden ser difícils d'entendre. Tot això pot provocar que els usuaris abandonin abans de finalitzar-lo o que contestin de qualsevol manera per acabar ràpid.

Per tal de millorar aquests aspectes, s'ha fet una anàlisi no exhaustiva de diverses possibilitats de gamificació: afegir vinyetes autoexplicatives en cada pregunta, afegir un fil lúdic al llarg del qüestionari, o trencar el model tradicional de qüestionari agafant les preguntes individualment i integrant-les en un joc. Els pros i contres de cada opció es desglossen en la taula següent:

	Pros	Contres	Cost/ complexitat
Afegir vinyetes en cada pregunta	<ul style="list-style-type: none"> - Ajuda a entendre les preguntes. - Ràpid: no allarga la durada del qüestionari. - S'integra en el qüestionari. 	<ul style="list-style-type: none"> - Menys lúdic. - Es requereix un dissenyador gràfic que dissenyi les 32 vinyetes amb coherència. - Es requereix un psicòleg que determini si les vinyetes són adequades per tal d'ajudar el pacient sense esbiaixar la seva resposta. 	<ul style="list-style-type: none"> - Cost elevat. - Complexitat tècnica baixa - Complexitat de realització elevada
Afegir un fil lúdic	<ul style="list-style-type: none"> - Manté l'interès - S'integra en el flux del qüestionari. - Permet establir fites. - No interfereix amb el qüestionari. 	<ul style="list-style-type: none"> - Pot distreure. - Relativament lent: allarga la durada del qüestionari depenent de l'activitat triada. - No ajuda a entendre les preguntes. 	<ul style="list-style-type: none"> - Cost i complexitat ajustables depenent dels mecanismes implementats.

Trencar el model de qüestionari	<ul style="list-style-type: none"> - Potencia molt la part lúdica - Pot permetre millorar la comprensió (exemple: joc situacional) 	<ul style="list-style-type: none"> - Lent: allarga la durada del qüestionari. - Alta complexitat. - Pot distreure i resultar confús. - El qüestionari perd importància 	<ul style="list-style-type: none"> - Cost i complexitat ajustables però habitualment molt elevats.
---------------------------------	--	--	---

Després d'analitzar els pros i contres, les limitacions i el cost de les diferents opcions i descartar per qüestions de cost de desenvolupament la possibilitat d'integrar en una estratègia conjunta dues o més de les possibilitats, s'ha optat per la segona opció. En efecte, és la que pot desenvolupar-se amb un cost més baix i la més viable per a l'estudiant a nivell autònom, a més de contribuir a mantenir l'interès de l'usuari i amenitzar el qüestionari. La finalitat de millorar l'enteniment de les preguntes requereix una anàlisi psicològica més exhaustiva i la feina d'un dissenyador gràfic, per la qual cosa s'ha deixat fora de l'abast del TFG. L'última opció presenta un cost massa elevat i un alt risc de distreure o confondre l'usuari per la seva gran complexitat.

Partint d'aquesta base, calia un joc divisible, per tal de poder integrar-lo entre les 32 preguntes, escalable, per mantenir l'interès, i amb un objectiu final, per tal d'evitar abandonaments en mig del qüestionari. A més, calia un joc senzill i que requerís poc temps i esforç, per tal de reduir les distraccions.

4.6.2 Mecanisme de gamificació escollit

Per tots els aspectes detallats, s'ha optat, finalment, pel joc de la imatge oculta. El joc consta de 32 oportunitats (una per pregunta) en què l'usuari ha d'endevinar la imatge oculta a partir dels seus fragments visibles, i conclou amb una pregunta final que enllaça totes les anteriors.

En respondre una pregunta del qüestionari, a l'usuari se li presenta una nova peça a l'atzar d'una imatge. Les imatges estan totes dividides en 9 peces. Si l'usuari no sap encara quin és el mot que representa la imatge amagada o respon incorrectament, el proper cop se li afegeix una nova peça a les anteriors. Es continua afegint peces fins que l'usuari esbrina el que amaga o s'arriben a mostrar les 9 peces que la componen, cas en què es mostra la imatge completa, es canvia d'imatge i es torna a començar amb una nova.

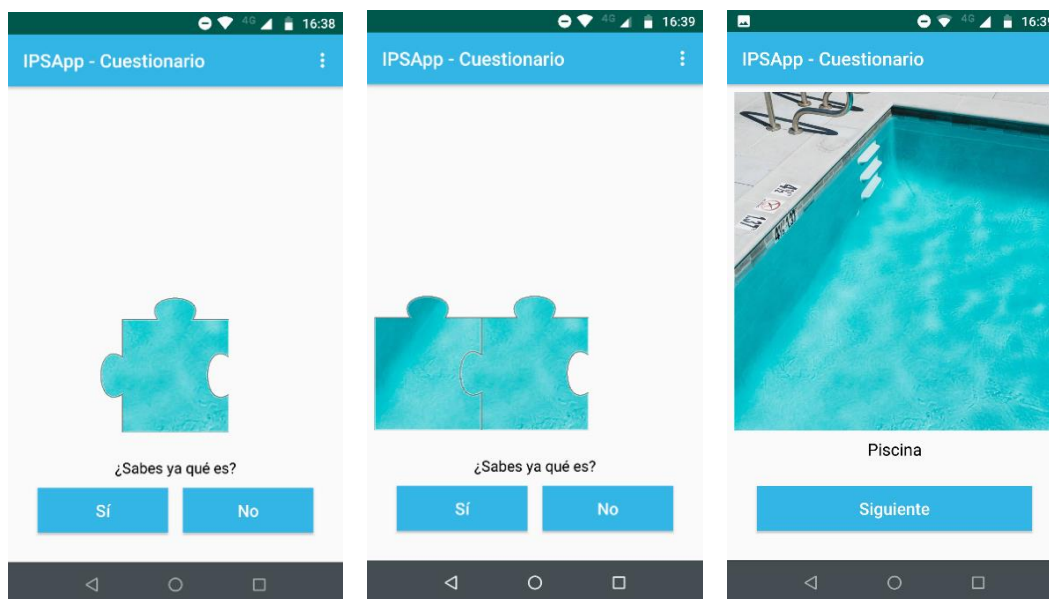


Figura 6. Pantalles del joc entre preguntes

Però el joc no s'acaba aquí. Després d'haver finalitzat el qüestionari i les 32 fases del joc, l'usuari accedeix a una pantalla que li mostra totes les imatges que ha encertat o arribat a completar i ha d'encertar quin és el concepte que les relaciona. Després, se l'informa de la resposta correcta, es dona el joc per conclòs i es procedeix a veure els resultats del qüestionari. Així, el resultat del joc i del qüestionari esdevenen una doble recompensa final.



Figura 7. Pantalla final del joc

En aquest joc, les imatges simbolitzen els nivells, que representen petites metes o recompenses integrades en el joc abans d'arribar a l'objectiu final, proporcionant així un sentit de progressió continu i una sensació d'assoliment cada cert temps, que fa que l'usuari mantingui l'atenció. Per una altra banda, el joc escollit és molt ràpid, probablement

més que les preguntes del qüestionari, de forma que no incrementa massa l'esforç necessari per respondre'l.

Malgrat el joc es podia haver deixat simplement en les 32 fases d'esbrinar la imatge oculta entre preguntes del qüestionari, s'ha volgut introduir la pregunta final per atorgar-li continuïtat i que l'usuari tingui un incentiu per encertar imatges i arribar al final del qüestionari. Representa el fil conductor del joc. A més, d'aquesta manera, es juga amb la possibilitat d'arribar a la "meta" amb un nombre d'imatges de 3 (divisió entera de les 32 fases del joc entre les 9 peces de cada imatge) a 32, la qual cosa s'empra com a mecanisme de recol·lecció i superació.

Actualment, l'app disposa de 3 conceptes diferents amb almenys 32 imatges associades a cadascun. Així, s'assegura que l'aplicació no es quedarà sense imatges si l'usuari les encerta totes a la primera. A més, en el cas més probable que no ho faci, encara quedarien imatges no vistes per l'usuari que poden aparèixer si torna a repetir el qüestionari en un futur amb el mateix concepte assignat. Aquestes imatges tenen llicència de compartir i modificar lliurement. En la versió actual s'han escollit de forma arbitrària, però es podrien substituir per imatges relacionades amb la temàtica del qüestionari. És possible augmentar el nombre d'imatges i de conceptes o substituir-ne alguna sense necessitat de modificar el codi.

Cal mencionar que el codi utilitzat per tal de dividir la imatge en peces amb forma de puzzle i per ensenyar el grid final s'ha adaptat del d'un joc per crear puzzles [5].

Cal destacar que també s'ha desenvolupat una versió no gamificada del joc per tal de poder analitzar els efectes de la gamificació. Es presenta en l'apartat 4.8.2.

4.7 Implementació

En aquest apartat es descriuen amb més detall les eines usades en la implementació de l'aplicació: mòduls, classes, funcions, etc.

4.7.1 Servidor

Com ja s'ha descrit prèviament, per a la implementació del servidor s'ha fet ús del microframework Flask de Python, incloses diverses extensions. A més, s'ha implementat Bootstrap pel disseny de la interfície gràfica i s'ha utilitzat una estructura de blueprints.

4.7.1.1 Bootstrap

Quant a l'aspecte i la funcionalitat base de l'aplicació, s'ha usat el framework de codi obert Bootstrap, que proporciona les plantilles utilitzades com a base dels fitxer HTML creats i

diverses eines de disseny, com formularis, botons o menús desplegable, basades en HTML i CSS, així com extensions de JavaScript addicionals.

L'aplicació web disposa de 2 parts diferenciades: les pantalles prèvies a l'inici de sessió i les de després. Tots els fitxers HTML que componen la primera part hereten de *layout_public.html* i mantenen una barra en la part superior que permet moure's entre la pàgina d'inici, la de presentació i la de login (o accedir a l'administració en cas de tenir la sessió iniciada). La segona part manté una barra en la part superior que permet tancar la sessió en qualsevol moment i compren els fitxers que hereten de *layout_admin.html*. Els 2 fitxers HTML mencionats són plantilles descarregades de la web oficial de Bootstrap juntament amb els fitxers de JavaScript i CSS necessaris, però modificades per tal que funcionin amb Flask i inicialitzin bootstrap i altres components com JQuery. Com que aquesta inicialització és comuna, s'ha implementat en un fitxer base del qual hereten els altres, anomenat *layout_bootstrap.html*.

4.7.1.2 Estructura de blueprints

Cal destacar l'estructura de blueprints utilitzada, que permet separar de forma lògica parts independents del servidor, la qual cosa millora la navegabilitat i facilita la localització de mòduls, la detecció d'errors i la gestió del servidor en general. S'han implementat 2 blueprints: un per a les gestions relacionades amb els usuaris de l'app web, i l'altre per a les relacionades amb el client. En el primer es guarden totes les funcions i plantilles referides a les pantalles que veu l'administrador o superadministrador dins la web i les funcions que pot desenvolupar, així com les plantilles dels correus que s'envien dins d'aquestes funcions. En el darrer es guarden les funcions que formen part de l'API REST i es duen a terme com a resposta a una petició del client, així com les plantilles dels correus que s'envien dins d'aquestes funcions. Finalment, les funcions de configuració i inicialització de l'app i les seves extensions, així com les plantilles informatives que es localitzen abans de l'inici de sessió en la web se situen fora de qualsevol blueprint.

A més, s'ha implementat el patró "Application Factory", per evitar tenir una variable app global i els problemes de referències circulars que sense cura pot ocasionar.

4.7.1.3 Flask-WTF

S'ha utilitzat aquesta extensió de Flask per crear els formularis de crear usuari o administrador, editar-lo i iniciar sessió. També per validar els formularis, a través del mètode *validate_on_submit* i l'ús del paràmetre *validators* en definir els seus camps. En els formularis de crear usuari i administrador es valida que s'hagin emplenat els camps d'email i contrasenya i que l'email introduït no estigui registrat. En els d'editar, es valida que s'hagi emplenat el correu electrònic i que les contrasenyes coincideixin.

4.7.1.4 Flask–Bootstrap

S'ha utilitzat principalment per fer compatible Bootstrap amb WTForms i Flask-WTF i poder així renderitzar formularis sencers en una sola ordre (*render_form*) o els seus camps de manera individual (*render_field*). En els formularis per editar un usuari o administrador, s'ha fixat el paràmetre *readonly* de la funció que renderitza el camp d'email a "readonly" per tal de deixar fixat el correu electrònic de l'usuari o administrador seleccionat.

```
{% block content %}
<form action="" method="post" class="form" role="form">
    {{ form.csrf_token }}
    {{ render_field(form.email, readonly='readonly') }}
    {{ render_field(form.password) }}
    {{ render_field(form.confirm_password) }}
    {{ render_field(form.active) }}
    {{ render_field(form.submit) }}
</form>
{% endblock %}
```

Figura 8. Fitxer HTML del formulari d'editar usuari

```
{%- extends "layout-admin.html" %}
{% from 'bootstrap/form.html' import render_form %}

{% block content %}
    {{ render_form(form) }}
{% endblock %}
```

Figura 9. Fitxer HTML del formulari d'afegir usuari

4.7.1.5 Flask–SQLAlchemy

Permet crear i mapejar la base de dades MySQL del servidor amb classes de Python. SQLAlchemy crea la base de dades a partir de l'estructura de les classes definides. Es detalla la creació de la base de dades en l'apartat "Model de dades".

4.7.1.6 Flask–Security

És dels mòduls de flask més utilitzats en l'aplicació. S'utilitza per limitar l'accés de certes pàgines de la web a usuaris amb certs rols o amb sessió iniciada, a través dels decoradors *@roles_accepted* i *@login_required* respectivament. A més, implementa un magatzem de dades d'usuaris i rols integrat amb SQLAlchemy que simplifica la seva creació, obtenció i eliminació. Tot això és possible gràcies a que les classes *User* i *Role* hereten, respectivament, de *UserMixin* i *RoleMixin* de Flask-Security. Per últim, es fa ús de la funció *utils.encrypt_password* per encriptar la contrasenya dels usuaris creats o editats abans de guardar-la a la base de dades.

Cal comentar que Flask-Security utilitza molts missatges automàtics i, malgrat trobar-se en moltes llengües, no conté les traduccions ni al català ni al castellà. En conseqüència, s'ha

hagut de traduir manualment el fitxer de missatges en anglès a ambdós idiomes i posar-los a la carpeta de traduccions de Flask-Security dins l'entorn virtual. Això, però, presenta un problema: si, per algun motiu, s'ha de tornar a instal·lar o actualitzar l'extensió, aquests fitxers desapareixerien i s'haurien de tornar a col·locar al lloc corresponent.

4.7.1.7 Flask-Mail

S'utilitza per enviar mails customitzats. Hi ha 3 casos en què s'envien correus electrònics al llarg de l'aplicació:

- **Correu de benvinguda:** quan es registra l'usuari dins l'aplicació, se li envia un correu de benvinguda amb la seva contrasenya.
- **Correu de canvi de contrasenya:** quan un administrador canvia la contrasenya d'algun usuari, se li envia un correu amb la nova contrasenya a l'usuari..
- **Correu de restabliment de contrasenya:** quan un usuari sol·licita restablir la seva contrasenya des de l'app mòbil, se li envia un correu amb una nova contrasenya.

Els 2 últims casos utilitzen la mateixa plantilla, només canvia el concepte del correu. Tots els mails estan compostos d'un mateix text traduït al català, castellà i anglès i s'envien des de l'adreça ipsappcontact@gmail.com. Per tal de crear les plantilles HTML dels correus s'ha descarregat una plantilla adaptativa [6] que fa ús d'HTML i Bootstrap i s'ha modificat bastant per adaptar-la a les necessitats de l'aplicació.

S'ha escollit utilitzar aquesta extensió en comptes dels correus automàtics que incorpora Flask-Security a causa de la necessitat de personalitzar el mail amb la contrasenya atorgada a l'usuari. A més, d'aquesta manera es pot escollir en quin moment s'envia el correu sense la limitació d'haver de realitzar una acció concreta.

El mòdul *emails* conté la funció *send_email*, amb la qual s'envien els mails. Per tal de fer-ho, es crea un objecte de la classe *Message* que conté el remitent, els destinataris, el subjecte, i el cos en format TXT i HTML, que es visualitzarà d'una manera o una altra en funció de com tinguin configurat l'email els destinataris. Cal destacar l'ús del mètode *attach* de la classe *Message* per tal d'incorporar la imatge del logo de l'aplicació en el correu. Els emails s'envien de manera asíncrona, de manera que la funció cridada finalitza i l'aplicació pot continuar el seu curs mentre l'enviament del mail segueix com a tasca en segon pla.

```

from .decorators import asyncfunc
from flask_mail import Message
from MobileTests.extensions import mail
from flask import current_app

@asyncfunc
def send_async_email(app, msg):
    with app.app_context():
        mail.send(msg)

def send_email(subject, sender, recipients, text_body, html_body, image='', extension=''):
    msg = Message(subject, sender=sender, recipients=recipients)
    msg.body = text_body
    msg.html = html_body
    app = current_app._get_current_object()
    if image!='':
        with app.open_resource('static/img/'+image+'.'+extension) as fp:
            msg.attach(image, 'image/'+extension, fp.read(), 'inline', headers = [['Content-ID', '<'+image+'>'],])
    send_async_email(app, msg)

```

Figura 10. Mòdul *emails*

```

from threading import Thread

def asyncfunc(f):
    def wrapper(*args, **kwargs):
        thr = Thread(target=f, args=args, kwargs=kwargs)
        thr.start()
        return wrapper

```

Figura 11. Mòdul *decorators*

Figura 12. Part en castellà del correu de benvinguda

4.7.1.8 Flask-Debugtoolbar

S'ha utilitzat Flask-Debugtoolbar a l'hora de programar l'aplicació per tal d'obtenir informació sobre l'execució del servidor: variables de sessió, peticions, temps d'execució, configuració, etc., i poder identificar i corregir errors. En activar el mode de desenvolupament de Flask a l'hora d'executar el programa la barra d'eines s'injecta automàticament en les respostes HTML.

4.7.1.9 Flask-BabelEx

Ha permès dotar l'aplicació de suport multilingüe. S'ha utilitzat la funció “_” en els fitxer HTML i en els scripts de Python per tal de marcar els texts a traduir. Pels camps dels formularis s'ha utilitzat la funció *lazy_gettext*, que tradueix el seu contingut al moment de rebre una petició, és a dir, quan realment s'ha d'utilitzar com a string en l'aplicació. Després, s'han extret tots els texts a traduir, s'han traduït amb el programa Poedit i s'han compilat per tal que la web es pogués mostrar en castellà (idioma per defecte), català i anglès.

S'ha utilitzat Flask-BabelEx en comptes de Flask-Babel, encara que el darrer es trobi en un estat més avançat en l'actualitat, perquè és l'extensió utilitzada per Flask-Security per realitzar les traduccions.

Cal destacar la funció encarregada de detectar la llengua del navegador d'aquell que es connecta a la web, definida en inicialitzar l'aplicació. S'ha creat una funció personalitzada perquè les usades en la documentació o altres versions trobades presenten un problema principal que s'ha volgut solventar: si l'usuari té configurada una llengua d'una regió concreta dins el navegador i l'aplicació suporta la mateixa llengua però d'una altra regió o la versió general, la pàgina surt en la llengua per defecte, en comptes de en la equivalent suportada, fins i tot utilitzant la funció *best_match*. Això succeïa, per exemple, si la llengua del navegador era “ca_ES”, ja que l'aplicació suporta “ca” i, per tant, es mostrava l'aplicació en castellà (llengua per defecte). Per això s'ha creat una funció que, en cas de trobar la llengua completa de l'usuari dins les llengües suportades per l'aplicació, agafa aquella i, en cas de no trobar-la, busca si troba alguna coincidència agafant només les 2 primeres lletres, abans de cridar la funció *best_match*.

```
@babel.localeselector
def get_locale():
    # no user locale stored. Deduce from browser accept_languages (custom logic)
    availablelanguages = current_app.config['LANGUAGES']
    for lang in request.accept_languages.values():
        if lang in availablelanguages:
            return lang
        if lang[:2] in availablelanguages:
            return lang[:2]
    return request.accept_languages.best_match(availablelanguages)
```

Figura 13. Funció “get_locale” de Flask-babelEx

4.7.1.10 Flask–Excel

Ha permès habilitar l'opció de descarregar en format XLS els resultats de tots els tests o les respostes d'un test concret dins l'aplicació web. Això permet obrir els fitxers amb Excel i facilita l'anàlisi i el seguiment de les dades dels usuaris per part dels administradors. La primera opció genera un fitxer amb 4 columnes associades a l'email de l'usuari, la data i hora del test realitzat, l'EB i el PB obtinguts. La segona genera un fitxer amb 4 columnes: el número de pregunta, la resposta escollida, la causa i el percentatge introduïts per l'usuari per a totes les preguntes del qüestionari. Així, aquest fitxer sempre té 32 files de dades i conté el correu de l'usuari i la data i hora d'enviament del qüestionari en el nom.

4.7.1.11 PyJWT

Per tal de gestionar la sessió en l'aplicació mòbil s'han usat *JSON Web Tokens* (JWT). Cada cop que s'inicia sessió en l'aplicació es genera un token d'aquest tipus en el servidor, el qual es codifica mitjançant la llibreria de Python PyJWT i s'envia i es guarda en la part de l'app mòbil. Cada cop que es requereix comprovar la validesa del token (en totes les peticions de l'API REST menys la d'inici de sessió) es descodifica a través de la mateixa llibreria.

4.7.1.12 Connexions amb l'API REST

Totes les connexions amb l'API REST per part del servidor es troben al mòdul *views.py* del blueprint *mobile*. S'utilitza la funció *jsonify* de flask per tal de convertir diccionaris en objectes JSON i enviar-los al client amb la informació necessària o un missatge informatiu, i la funció *get_json* del mòdul *request* de flask per tal d'obtenir l'objecte JSON provinent del client. En cas de retornar un missatge d'error, es fa juntament amb el codi d'error corresponent.

4.7.2 Client

Com s'ha mencionat prèviament, pel desenvolupament de l'aplicació mòbil s'ha utilitzat el software de codi obert Android Studio. En l'apartat present s'expliquen diversos aspectes del desenvolupament de l'aplicació mòbil, les activitats creades, altres classes i *layouts*, particularitats de la metodologia i tecnologia utilitzada, etc. No s'explica la creació de la base de dades SQLite ja que es pot trobar en l'apartat 4.4.2. En l'annex 3 pot consultar-se un diagrama de les classes creades.

4.7.2.1 Classes de gestió

Hi ha diverses classes que no hereten de cap activitat i s'utilitzen per crear objectes o gestionar part de l'aplicació. Aquestes són:

- **QuestContract:** conté l'estructura de la base de dades del client. Inclou 3 classes que fan referència a les 3 taules de la base de dades, amb atributs estàtics finals que indiquen el seu nom i el de les seves columnes.
- **DatabaseHelper:** hereta de `SQLiteOpenHelper` i conté les funcions de creació i actualització a noves versions de la base de dades, així com altres funcions d'inserció, consulta o eliminació de dades que s'utilitzen en diverses activitats.
- **SessionManager:** és la classe encarregada de gestionar la sessió. Utilitza la interfície `SharedPreferences` per guardar i recuperar dades globals de l'aplicació. Així, quan un usuari inicia sessió es guarda el seu correu electrònic i el token que se li ha assignat, juntament amb una clau que indica que la sessió està iniciada. La classe conté 3 mètodes que retornen cadascun dels paràmetres guardats i una funció per tancar sessió, que esborra tots els valors guardats (les claus tornen al valor que tenen per defecte) i redirecciona l'usuari a la pantalla de login.
- **AlertDialogManager:** conté un mètode que permet crear i mostrar un quadre de diàleg convencional de tipus `AlertDialog` com a pop-up en la pantalla. Aquest s'utilitza en diverses ocasions al llarg de l'aplicació per advertir d'errors d'incomplència, d'incompatibilitat, del servidor i d'altres, així com per informar que s'ha enviat un mail quan l'usuari introdueix l'email per restablir la seva contrasenya. El mètode conté diversos paràmetres: el context de l'aplicació, el títol del diàleg, el seu missatge i un booleà que indica si es tracta d'un diàleg d'error (*false*), cas en què es mostra una creu vermella, o de tasca completada amb èxit (*true*), cas en què es mostra un tick verd. Els diàlegs creats amb aquest mètode presenten un únic botó de "ok", el qual tanca el diàleg.
- **AlertDialogPuzzle:** hereta de `AppCompatActivity` i conté el mètode que crea els diàlegs en què l'usuari introdueix la seva resposta per a una imatge del joc. Cal un mètode específic perquè requereix un `EditText`, el qual es troba en el fitxer `dialog_puzzle.xml`, on l'usuari pugui escriure la seva resposta. A més, cal definir 2 botons i la seva funcionalitat: el botó de continuar, encarregat d'obrir un altre diàleg d'alerta diferent que indiqui si la resposta és correcta o no, i el botó de cancel·lar, que ha de tancar el diàleg.

```
AlertDialog.Builder builder = new AlertDialog.Builder(getActivity());

LayoutInflater inflater = getActivity().getLayoutInflater();
View view = inflater.inflate(R.layout.dialog_puzzle, root: null);

builder.setView(view)
```

Figura 14. Integració del *layout* en l'*AlertDialog*

- **ImageAdapter:** hereta de `BaseAdapter` i s'empra al final del joc per emplenar la *grid* on se situen les imatges del joc encertades o completament desbloquejades per l'usuari. S'encarrega de crear un `ImageView` per a cadascuna de les imatges que si li han passat, la qual cosa es fa a través de l'element `GridImageView` del fitxer `grid_element.xml`. A més, les redimensiona a una mida fixa i col·loca dins del *grid*, que també té una mida fixa i dins del qual es pot fer scrolling vertical.
- **PuzzlePiece:** hereta de `android.support.v7.widget.AppCompatImageView` i conté diversos atributs referents a les coordenades x i y de la peça d'una imatge del joc, la seva amplada i la seva alçada.

A més de les 2 classes mencionades per crear quadres de diàleg, a llarg de l'aplicació es creen i mostren diàlegs d'alerta directament en algunes activitats. Destaca el que apareix després que l'usuari hagi introduït una resposta en el joc, el qual indica si aquesta és correcta o no. Es tracta d'un diàleg que no desapareix si se'n toca la zona exterior, ja que té posada la propietat *setCancelable* a *false*. Així, només se'n surt si es clica el botó de continuar. Si s'ha encertat la imatge, apareix un altre *AlertDialog* que segueix la mateixa dinàmica i que porta a veure la imatge completa. També és d'aquest tipus el diàleg que apareix en finalitzar l'última fase del joc d'endevinar imatges i que permet enviar el qüestionari i el que indica si s'ha encertat o no el concepte final i que porta a veure els resultats.

4.7.2.2 Activitats

Hi ha una classe de Java i un fitxer XML per a cada activitat. La classe de Java conté la part funcional del codi, mentre que el fitxer XML en conté el *layout* o part gràfica. Totes les classes que corresponen a activitats hereten d'*AppCompatActivity*. L'aplicació està configurada per només ser utilitzada amb el mòbil en mode vertical.

- **SplashScreenActivity:** Correspon a la primera pantalla que apareix en obrir l'aplicació, que es manté durant un temps establert en 2 segons. L'activitat es connecta a *SessionManager* per saber si la sessió està iniciada, cas en què es valida el token a través d'una connexió amb el servidor. Si no es rep cap resposta del servidor en els 2 segons establerts, la petició espera un màxim de 7 segons mantenint la pantalla de splash visible, aleshores es genera un error, es tanca la sessió de l'usuari i se'l redirecciona a la pantalla d'inici de sessió. En cas que no hi hagi cap sessió iniciada, es redirecciona l'usuari a la pantalla de login directament passats els 2 segons.
- **RegisterActivity:** Conté 2 camps perquè l'usuari introdueixi el seu nom i sexe i ho guarda a la base de dades.
- **PasswordActivity:** Conté 2 camps perquè l'usuari introdueixi 2 cops la contrasenya que vol utilitzar i es connecta amb el servidor per guardar la nova contrasenya.
- **ForgotPasswordActivity:** Conté un camp perquè l'usuari introdueixi el seu correu i es connecta amb el servidor per tal de canviar la contrasenya de l'usuari i enviar-li un correu amb la nova contrasenya.
- **LoginActivity:** Conté 2 camps perquè l'usuari introdueixi el seu email i la seva contrasenya i es connecta amb el servidor per tal d'iniciar sessió si tots els paràmetres introduïts són correctes.
- **MainActivity:** Mostra un text amb el nom de l'usuari extret de la base de dades i un text i un botó diferent en funció de l'estat de l'aplicació. El botó porta a alguna fase del qüestionari o del joc.
- **InstructionsActivity, Instructions2Activity, Instructions3Activity:** Mostren les 3 pantalles d'instruccions de l'aplicació, una darrere de l'altra, excepte si es

pressiona el botó de saltar instruccions en alguna de les 2 primeres pantalles, cas en què s'accedeix directament a l'activitat que tocaria després de la 3a pantalla.

- **QuestionActivity:** S'ha emprat una única classe i un únic *layout* per a les 32 preguntes del qüestionari. En iniciar una activitat corresponent a la classe pregunta es passa informació que indica el número de pregunta que toca respondre. La classe rep el paràmetre, el fixa al *TextView* associat al número de pregunta, busca en el fitxer de strings quina és la pregunta que hi correspon i la fixa en la vista adequada. En prémer continuar, si no s'ha deixat cap camp buit, es guarda l'opció, la causa i el percentatge en la base de dades.
- **GameActivity, ImageActivity, EndActivity, Image2Activity:** Conjunt de pantalles associades al joc. També hi ha una única classe i un únic *layout* per pantalla. El seu mecanisme de funcionament s'explica en l'apartat "Joc".
- **ResultsActivity:** Agafa l'EB i el PB obtinguts pel qüestionari actual de la base de dades, calcula el resultat i fixa les vistes amb els textos corresponents.

4.7.2.3 Extres

Els extras són molt útils com a mecanisme per passar paràmetres al llarg de l'aplicació sense haver de guardar-los de manera global i ocupar espai innecessari en la memòria. S'utilitzen en moltes ocasions al llarg de l'aplicació:

- 1) Des de l'activitat principal (*MainActivity*) es passa un paràmetre referent a la propera pregunta que toca respondre si es va a *QuestionActivity* o a l'última pregunta resposta si s'accedeix a *GameActivity*.
- 2) De pregunta a joc es passa el número de l'última pregunta resposta i de joc a pregunta es passa el de la pregunta següent.
- 3) Si s'encerta una imatge, es passa el nom de la imatge i el de la resposta correcta a *ImageActivity*, així com el número de l'última pregunta resposta. Des d'aquesta activitat es passa el número de la propera pregunta a *QuestionActivity*.
- 4) En accedir al joc, es passa un extra corresponent a si és necessari afegir-li una peça a la imatge o no. Si l'intent prové de l'activitat *QuestionActivity*, caldrà afegir-la i per tant l'extra serà *true*. Altrament (l'intent prové de la pantalla principal i es reprèn l'aplicació per on estava), l'extra serà *false* i no s'afegirà cap peça.
- 5) Quan es fa clic al botó d'instruccions del menú desplegable de la barra superior de l'aplicació, es passa un paràmetre referent a si s'ha fet clic des d'una pregunta del qüestionari o des del joc. En el primer cas, es passa també el número de la pregunta des de la qual s'ha clicat i en el segon el de la pregunta anterior. Si es clica des de la pantalla principal es passa, en comptes d'això, el número 200. El mateix extra es passa a les 2 pantalles d'instruccions següents.
- 6) Per passar des del *grid* final la imatge a mostrar per *Image2Activity* en clicar alguna de les anteriors i la resposta associada a aquesta.

4.7.2.4 Joc

Totes les imatges utilitzades en el joc estan guardades en la carpeta *img* dins la carpeta *assets*. Quan s'inicialitza la base de dades del client, es creen les taules *conceptImages*, que conté totes les imatges disponibles i el seu concepte associat, i *concepts*, que conté tots els conceptes utilitzables. Quan es comença un qüestionari, se selecciona un concepte a l'atzar de la darrera taula i una imatge aleatòria associada a aquest concepte de la primera, els quals es guarden en els paràmetres *CurrentConcept* i *CurrentImage* de la taula *users* respectivament.

Cada cop que s'accedeix al joc (activitat *GameActivity*), es recupera la imatge, es fixa en la vista corresponent i es divideix en 9 peces en forma de puzzle mitjançant el mètode *splitImage*. Aleshores es converteix el string *SeenPieces* de taula d'usuaris en una llista d'índexs de peces vistes d'aquella imatge mitjançant el mètode *split* i dividint la cadena pel caràcter “;”. Es mostren totes les peces corresponents a aquests índexs, juntament amb una peça addicional aleatòria si l'extra corresponent així ho indica (veure punt 4 de l'apartat d'extres), l'índex de la qual s'afegeix com a string al final de *seenPieces*, seguida d'un punt i coma.

Quan l'usuari encerta la imatge o desbloqueja les 9 peces, el paràmetre *seenPieces* i *CurrentImage* s'actualitzen a null i s'afegeix l'id de la imatge com a string acabat en punt i coma al final de *SeenImages* per tal que no torni a sortir. Després, es procedeix a mostrar la imatge (activitat *ImageActivity*). El proper cop que s'accedeixi al joc es fixarà una nova imatge associada al mateix concepte. Se separa el paràmetre *SeenImages* pel caràcter “;” i s'obté una llista que conté els identificadors de les imatges ja vistes, se selecciona una imatge aleatòria de *ConceptImages* i es repeteix el procés fins que el seu id no es troba en la llista d'imatges vistes, aleshores es fixa en la vista corresponent.

Després de l'última fase del joc d'endevinar imatges, s'accedeix a *EndActivity*, que recupera les imatges de *seenImages* i mostra totes les imatges encertades o completament desbloquejades en un *grid*, fent ús de la classe *ImageAdapter*, com ja s'ha explicat.

Cal destacar el mètode que comprova si la resposta introduïda per l'usuari dins del joc és correcta. Per a cada imatge, en el fitxer que conté els strings a *resources*, s'ha posat com a clau el nom de la imatge juntament amb la seva extensió (substituint el punt per una barra baixa ja que el primer donava problemes) i com a valor totes les respostes correctes separades pel caràcter “|”. A l'hora de recuperar la resposta, s'agafa el nom de la imatge actual (inclosa l'extensió) de la base de dades, se substitueix el punt que precedeix l'extensió per una barra baixa i es busca el valor que correspon a aquesta clau en el fitxer de strings. S'eliminen tots els signes diacrítics de la cadena obtinguda utilitzant el mètode *normalize* de la classe *Normalizer*, que els separa de la resta de caràcters, i el mètode *replaceAll*, que els elimina. Després, es passa el nou string a lletra minúscula i se separa

pel caràcter "]" utilitzant el mètode *split*. Finalment, s'eliminen també els signes diacrítics de la resposta de l'usuari i es passa a minúscula abans de comprovar si algun dels elements de la llista obtinguda la conté.

```
public void confirm(String answer) {
    updatePuzzleState();
    SQLiteDatabase db = admin.getWritableDatabase();
    String query2 = "SELECT * FROM " + UsersTable.TABLE_NAME + " INNER JOIN " +
        ConceptImagesTable.TABLE_NAME + " ON " + ConceptImagesTable.COL_ID_CONCEPT_IMAGE +
        " = " + UsersTable.COL_CURRENT_IMAGE + " WHERE " + UsersTable.COL_EMAIL + " = ?";
    String[] selectionArgs = {email};
    Cursor cursor = db.rawQuery(query2, selectionArgs);
    if (cursor.moveToFirst() && cursor.getCount() > 0) {
        final String image = cursor.getString(cursor.getColumnIndex(ConceptImagesTable.COL_IMAGE_NAME));
        final String _correctAnswer = getString(getResources().getIdentifier(image.replace
            (target: ".", replacement: "_"), defType: "string", getPackageName()));
        final String correctAnswer = _correctAnswer.split( regex: "\\|"[0]; //correct version to show
        String nCorrectAnswer = (Normalizer.normalize(_correctAnswer, Normalizer.Form.NFD));
        nCorrectAnswer = nCorrectAnswer.replaceAll( regex: "[^\\p{ASCII}]", replacement: "").toLowerCase();
        List<String> correctAnswers = Arrays.asList(nCorrectAnswer.split( regex: "\\|")); //accepted versions to compare
        String nAnswer = (Normalizer.normalize(answer, Normalizer.Form.NFD));
        nAnswer = nAnswer.replaceAll( regex: "[^\\p{ASCII}]", replacement: "").toLowerCase();
        if (correctAnswers.contains(nAnswer)) {
            //answer is correct
        }
    }
}
```

Figura 15. Part del mòdul que comprova si la resposta és correcta

Quant a la comprovació del concepte, el procediment és el mateix, només canvia la clau utilitzada en el fitxer de strings, que equival al nom que se li ha assignat al concepte en la base de dades.

4.7.2.5 Menú desplegable

En la majoria d'activitats hi ha un menú desplegable a la part dreta de la barra superior de la pantalla. Aquest menú se situa en un *layout* a part anomenat *drowdown_menu.xml* i permet veure les instruccions o tancar la sessió. El botó d'instruccions només està disponible en la pantalla principal (veure figura 14), en les preguntes del qüestionari i les pantalles que mostren el joc, ja que són els 3 llocs on té més sentit que l'usuari vulgui recordar-les. En les pantalles prèvies a la principal, com que són ocasionals, només es permet tancar la sessió, i en les de després d'acabar el qüestionari i l'última fase del joc no apareix el menú, ja que es pretén evitar que l'usuari abandoni l'aplicació aleshores, en quedar molt poc per finalitzar completament.

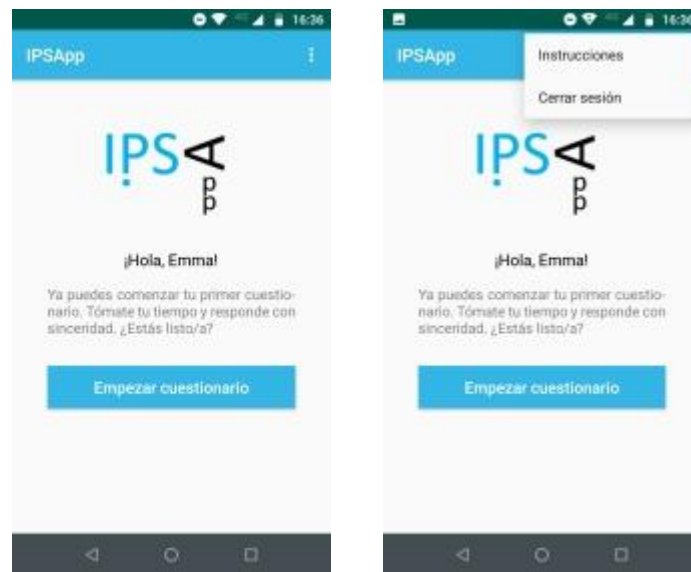


Figura 16. Menú desplegable en la pantalla principal

4.7.2.6 Scrolling vertical

Hi ha diverses activitats que consten fonamentalment de text distribuït al llarg de la pantalla: són les 3 activitats d'instruccions, la de les preguntes del qüestionari, la de resultats i la pantalla principal. Per fer que el *layout* s'adapti a dispositius de diferents mides, especialment aquells més petits, s'ha inclòs un *ScrollView*, que permet fer scrolling vertical si l'aplicació ho necessita. En aquells dispositius de mida estàndard o més grans l'aplicació no ho requereix i per tant el *layout* cap en la pantalla sense haver de fer scrolling. En la resta de pantalles no és necessari ja que inclouen majoritàriament o bé imatges que s'adapten a la mida del dispositiu o bé camps d'edició de text que fan el mateix, de manera que les dimensions no són un problema.

4.7.2.7 Suport multilingüe

A *resources* hi ha un fitxer de strings per a cada llengua que suporta l'aplicació: català, castellà i anglès. El fitxer conté una clau i un valor per a cada string que s'ha d'utilitzar en l'aplicació. Tots els strings estan posats amb la mateixa clau en els diferents fitxers de string i traduïts a la llengua pertinent. Quan s'han de mostrar a l'aplicació, es criden per la seva clau i Android els mostra en la llengua que l'usuari té configurada en el seu dispositiu mòbil. Si aquesta llengua no està disponible en l'aplicació, es mostraria l'idioma per defecte, que és el castellà.

4.7.2.8 Connexions amb l'API REST

Per tal de realitzar totes les connexions amb el servidor, s'utilitza la llibreria *Volley* d'Android.

En primer lloc, es crea un objecte de la classe *JsonObjectRequest*, al qual se li passa la URL del servidor al qual s'ha de connectar, el mètode amb què ho ha de fer (en l'aplicació presentada totes les connexions són GET o POST), els paràmetres a enviar amb la petició i 2 objectes de classe *Response.Listener<JSONObject>* i *Response.ErrorListener*, dels qual cal sobreescrivir el mètode *onResponse* i *onErrorResponse* respectivament per tal d'indicar que ha de fer l'aplicació en cada cas. També, en les peticions realitzades dins d'una sessió (enviar qüestionari, canviar contrasenya i validar token en obrir l'aplicació amb la sessió iniciada) se sobreescriví el mètode *Map<String, String> getHeaders* de la classe *JsonObjectRequest* per tal d'enviar com a *header* el token de sessió guardat i generar un error en cas que no sigui vàlid.

Es modifica la petició per establir un temps màxim d'espera de 7 segons abans de generar un error si no s'ha rebut resposta. Després es crea una nova cua de peticions de *Volley* i s'hi afegeix la petició creada.

```
int MY_SOCKET_TIMEOUT_MS = 7000;
JsonObjectRequest.setRetryPolicy(new DefaultRetryPolicy(
    MY_SOCKET_TIMEOUT_MS,
    DefaultRetryPolicy.DEFAULT_MAX_RETRIES,
    DefaultRetryPolicy.DEFAULT_BACKOFF_MULT));
RequestQueue requestQueue = Volley.newRequestQueue( context, this);
requestQueue.add(jsonObjectRequest);
```

Figura 17. Establiment del temps d'espera i enviament de la petició a través de *Volley*

4.8 Versions de l'aplicació mòbil

S'han realitzat dues versions de l'aplicació mòbil, una gamificada i una no gamificada, per tal de poder analitzar l'efecte de la gamificació sobre l'experiència d'usuari.

4.8.1 Versió gamificada

S'ha desenvolupat una versió del qüestionari IPSAQ que incorpora elements de joc (gamificació) per tal de millorar la disposició dels pacients a realitzar-lo.

Aquesta versió incorpora, òbviament, el qüestionari i, per això, en bona part, pot considerar-se una ampliació respecte a la versió no gamificada. El joc també ha imposat algun canvi d'estructura o flux, com es comentarà posteriorment.

Els detalls d'aquesta versió (com casos d'ús, tecnologia, flux, etc.) són els que s'exposen bàsicament en aquest projecte. Per aquest motiu, a continuació es desenvoluparan les diferències que presenta la versió no gamificada.

4.8.2 Versió no gamificada

S'ha desenvolupat una versió no gamificada de l'aplicació per tal de comparar la satisfacció i opinió dels usuaris respecte a l'altra versió i verificar, així, els efectes de la gamificació. La versió no gamificada de l'app presenta diverses singularitats. A diferència de la versió gamificada, en aquesta versió bàsica el flux del qüestionari no és necessàriament lineal, sinó que hi ha un menú des del qual es pot accedir a la pregunta que es desitja en cada moment. Des de qualsevol pregunta, es pot anar a l'anterior o a la següent clicant els botons corresponents i en un flux circular, és a dir, accedint a la pregunta 1 després de la 32 o a la 32 en clicar anterior des de la 1.

Una altra distinció d'aquesta versió es que es pot deixar la pregunta sense contestar o a mitges i canviar de pantalla, guardant els canvis i no mostrant cap missatge d'error. Es pot tornar al menú de preguntes en qualsevol moment fent clic al botó d'enrere del dispositiu.

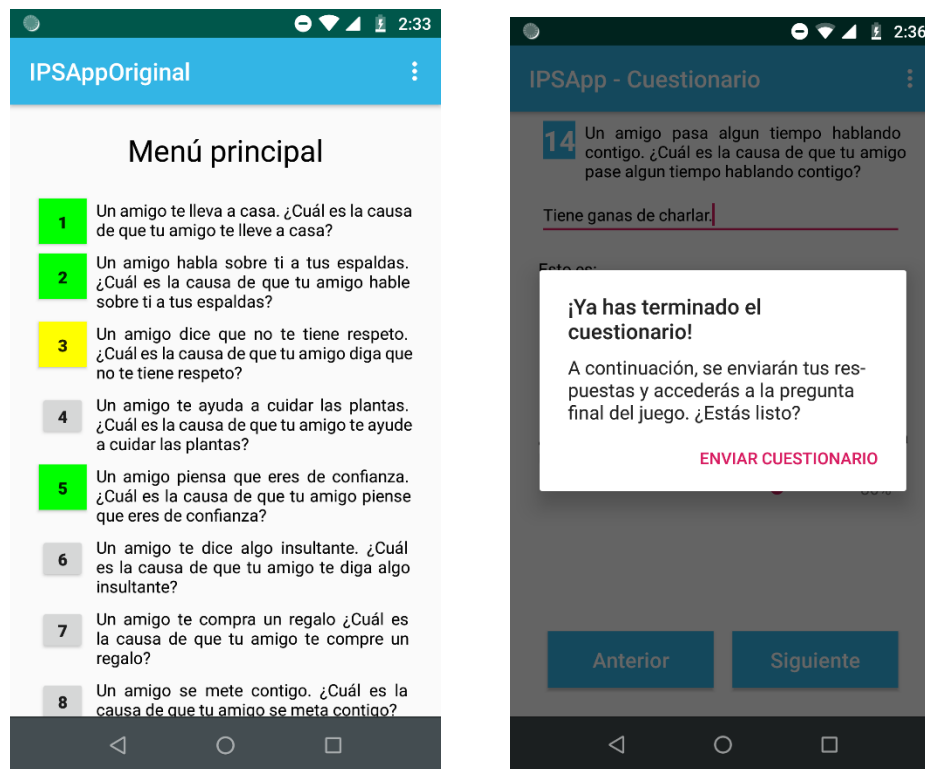


Figura 18. Versió sense gamificar de l'aplicació

El botó que permet accedir a les preguntes des del menú es mostra en colors diferents segons l'estat de la pregunta: si està completada es mostra en verd, si està a mitges en groc i si està buida manté el color gris original.

Després de respondre l'última pregunta, en intentar abandonar la pantalla realitzant qualsevol acció apareix una finestra en pantalla (no cancel·lable) que indica que s'ha finalitzat el qüestionari i permet enviar-lo. En cas de tancar l'aplicació sense haver-lo enviat,

en el proper accés apareixerà un missatge des de la pantalla principal indicant que s'ha d'enviar i un botó per fer-ho, a més de l'opció d'enviar activada des del menú desplegable de la part superior dreta de la pantalla.

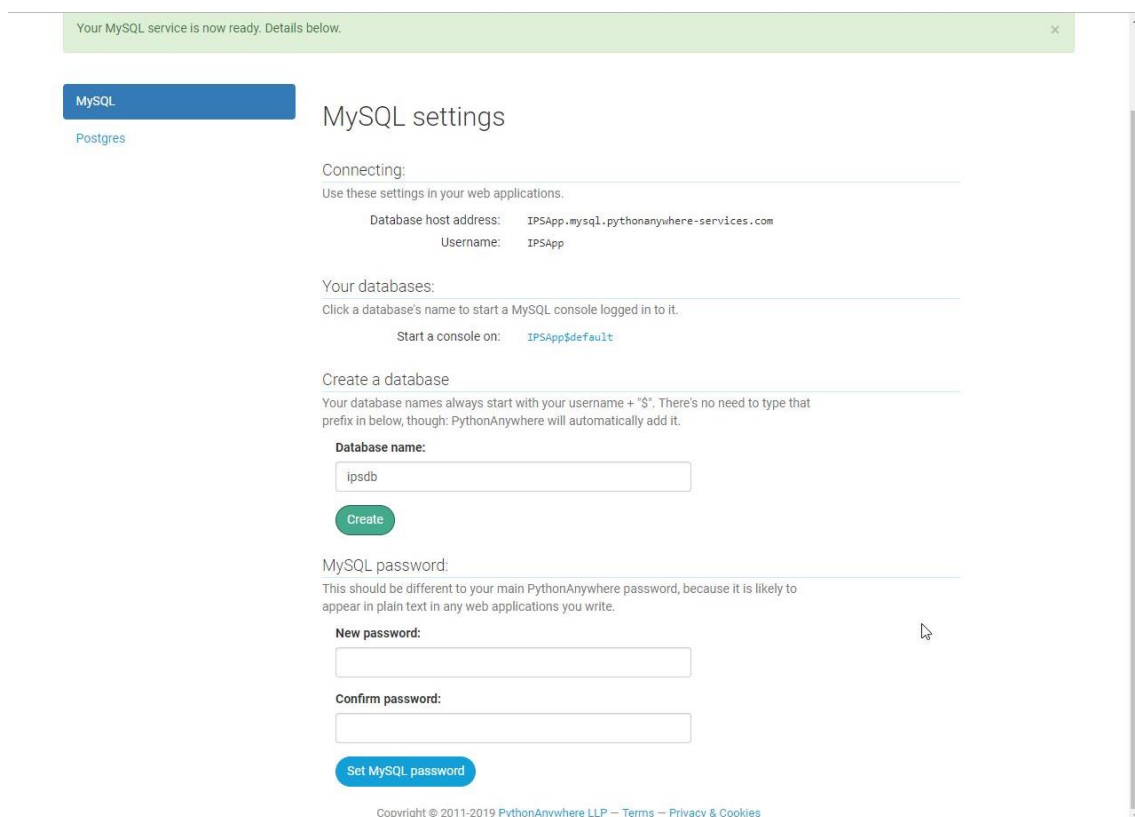
Les instruccions d'aquesta versió també són diferents, en indicar tot el que s'ha explicat prèviament.

Es va eliminar el menú de preguntes en la versió gamificada perquè d'aquesta manera el camí estigués més guiat, hi hagués un sentit de progressió major i les preguntes s'assimilessin a fases del joc.

5 Desplegament

Un cop finalitzada la programació de les 2 parts de l'aplicació, s'ha procedit a desplegar l'aplicació web en un servidor extern per tal de permetre accés des de qualsevol punt d'Internet. S'ha trobat el servei d'allotjament web gratuït PythonAnywhere, que proporciona totes les eines necessàries per fer-ho.

En primer lloc, s'ha configurat la base de dades MySQL que ha d'utilitzar el servidor per emmagatzemar les dades. S'ha creat una base de dades amb el nom `ipsdb` i un prefix automàtic i s'ha escollit una contrasenya. L'usuari correspon al del compte de PythonAnywhere (IPSAApp).



Your MySQL service is now ready. Details below.

MySQL
Postgres

MySQL settings

Connecting:
Use these settings in your web applications.

Database host address: `IPSAApp.mysql.pythonanywhere-services.com`
Username: `IPSAApp`

Your databases:
Click a database's name to start a MySQL console logged in to it.

Start a console on: `IPSAApp$default`

Create a database
Your database names always start with your username + '\$'. There's no need to type that prefix in below, though: PythonAnywhere will automatically add it.

Database name:

Create

MySQL password:
This should be different to your main PythonAnywhere password, because it is likely to appear in plain text in any web applications you write.

New password:

Confirm password:

Set MySQL password

Copyright © 2011-2019 PythonAnywhere LLP — Terms — Privacy & Cookies

Figura 19. Configuració base de dades MySQL

A continuació, s'ha canviat l'URI de la bases de dades dins del fitxer `config` de l'aplicació d'acord a la creada. Després, s'ha decidit quins fitxers de l'aplicació es volien penjar i s'ha esborrat la resta de la còpia a pujar (l'encarregat d'executar l'app `-runserver-`, els de cache `-pycache-` i l'entorn virtual) i, finalment, s'ha penjat la carpeta comprimida que conté els fitxers de l'app web i s'ha descomprimit fent un `unzip` des de la consola.

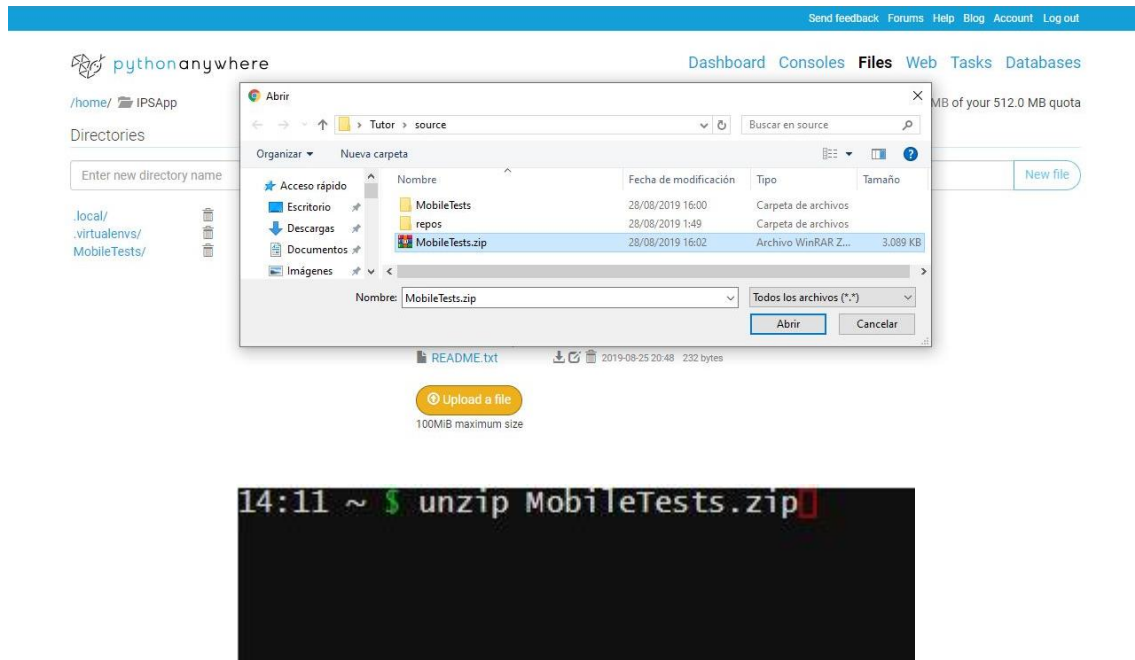


Figura 20. Desplegament dels fitxers de l'app web al servidor

Posteriorment, s'ha creat i configurat l'entorn virtual de l'aplicació, que utilitza Python 3.7. S'han instal·lat tots els paquets de Python necessaris des del fitxer *requirements.txt*, que conté el nom i la versió de tots els paquets que cal instal·lar addicionalment per executar l'app correctament.

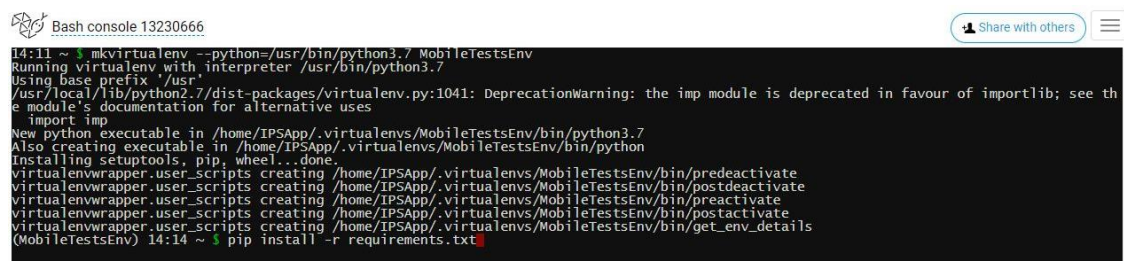


Figura 21. Creació i configuració de l'entorn virtual

En haver de crear de nou l'entorn virtual, s'han perdut les traduccions manuals de Flask-Security al català i castellà, així que s'ha pujat la part de la carpeta *translations*, que les conté, i s'ha descomprimit directament a la ruta corresponent de l'entorn virtual.

```

(MobileTestsEnv) 14:56 ~ $ ls
MobileTests MobileTests.zip README.txt requirements.txt translations.zip
(MobileTestsEnv) 14:56 ~ $ unzip translations.zip -d /home/IPSApp/.virtualenvs/MobileTestsEnv/lib/python3.7/site-packages/flask_security/t
translations
Archive: translations.zip
  creating: /home/IPSApp/.virtualenvs/MobileTestsEnv/lib/python3.7/site-packages/flask_security/translations/es_ES/
  creating: /home/IPSApp/.virtualenvs/MobileTestsEnv/lib/python3.7/site-packages/flask_security/translations/es_ES/LC_MESSAGES/
  inflating: /home/IPSApp/.virtualenvs/MobileTestsEnv/lib/python3.7/site-packages/flask_security/translations/es_ES/LC_MESSAGES/Flask_secu
rity.mo
  inflating: /home/IPSApp/.virtualenvs/MobileTestsEnv/lib/python3.7/site-packages/flask_security/translations/es_ES/LC_MESSAGES/Flask_secu
rity.po
  creating: /home/IPSApp/.virtualenvs/MobileTestsEnv/lib/python3.7/site-packages/flask_security/translations/ca/
  creating: /home/IPSApp/.virtualenvs/MobileTestsEnv/lib/python3.7/site-packages/flask_security/translations/ca/LC_MESSAGES/
  inflating: /home/IPSApp/.virtualenvs/MobileTestsEnv/lib/python3.7/site-packages/flask_security/translations/ca/LC_MESSAGES/Flask_securit
y.mo
  inflating: /home/IPSApp/.virtualenvs/MobileTestsEnv/lib/python3.7/site-packages/flask_security/translations/ca/LC_MESSAGES/Flask_securit
y.po
(MobileTestsEnv) 14:58 ~ $

```

Figura 22. Addició de les traduccions al català i castellà de Flask-Security

Finalment, s'ha creat l'aplicació web, introduint la ubicació del directori de treball i l'entorn virtual i configurant el fitxer *wsgi*, encarregat d'executar l'aplicació, per tal que faci ús de la funció *create_app* del mòdul *__init__.py*. PythonAnywhere obliga a guardar-la en una variable anomenada *application*.

Want some more data? [Paying accounts](#) get pretty charts :-)

Code:

What your site is running.

Source code: [Enter the path to your web app source code](#)

Working directory: [/home/IPSApp/](#) [Go to directory](#)

WSGI configuration file: [/var/www/ipsapp.pythonanywhere.com_wsgi.py](#)

Python version: 3.7

Virtualenv:

Use a virtualenv to get different versions of flask, django etc from our default system ones. [More info here](#). You need to **Reload your web app** to activate it; NB - will do nothing if the virtualenv does not exist.

[/home/IPSApp/.virtualenvs/MobileTestsEnv](#)

[Start a console in this virtualenv](#)

Log files:

The first place to look if something goes wrong.

Access log: [ipsapp.pythonanywhere.com.access.log](#)

Error log: [ipsapp.pythonanywhere.com.error.log](#)

Server log: [ipsapp.pythonanywhere.com.server.log](#)

Log files are periodically rotated. You can find old logs here: [/var/log](#)

Static files:

Files that aren't dynamically generated by your code, like CSS, JavaScript or uploaded files, can be served much faster straight off the disk if you specify them here. You need to **Reload your web app** to activate any changes you make to the mappings below.

URL	Directory	Delete
Enter URL	Enter path	

Security:

An HTTPS certificate is needed so that people can access your site securely. We automatically provide a certificate for [IPSApp.pythonanywhere.com](#).

HTTPS certificate: [Automatically provided for this hostname](#)

```

61
62 import sys
63
64 ## The "/home/IPSApp" below specifies your home
65 ## directory -- the rest should be the directory you uploaded your Flask
66 ## code to underneath the home directory. So if you just ran
67 ## "git clone git@github.com:myusername/myproject.git"
68 ## ...or uploaded files to the directory "myproject", then you should
69 ## specify "/home/IPSApp/myproject"
70 path = '/home/IPSApp/MobileTests'
71 if path not in sys.path:
72     sys.path.append(path)
73
74 from MobileTests import create_app # noqa
75 application = create_app('config.py')

```

Figura 23. Configuració del servidor

D'aquesta manera, ja es té l'app web desplegada en un servidor extern, accessible des de l'adreça <http://www.ipsapp.pythonanywhere.com>.

S'ha detectat, però, un problema: al cap de 3 minuts d'inactivitat, la base de dades MySQL tancava les connexions i la propera crida donava error. Si es tirava enrere i es tornava a intentar, però, ja funcionava correctament. Es va aplicar la solució que els de PythonAnywhere proposaven, que implicava modificar el fitxer de configuració de l'aplicació per tal que SQLAlchemy mantingués menys temps les connexions, però, malgrat millorar el problema, no es va solucionar. Finalment, s'ha pogut arreglar el problema afegint la línia `SQLALCHEMY_ENGINE_OPTIONS = {'pool_pre_ping': True}` en el fitxer de configuració, la qual comprova abans de connectar-se a la base de dades MySQL si la connexió segueix activa.

El codi font de l'aplicació i l'APK s'han pujat a GitHub. El servidor s'ha desplegat a PythonAnywhere. La seva localització es detalla en l'annex 6.

6 Proves

6.1 Proves unitàries del servidor

A l'hora de programar el servidor, s'han anat realitzant proves en local després d'implementar cada funcionalitat per tal de comprovar el seu correcte funcionament. Això s'ha fet a través del programari Postman, que proporciona un client HTTP per l'API creada i permet enviar i rebre peticions amb els paràmetres adequats. D'aquesta manera, s'ha pogut comprovar que totes les rutes exercien la funció esperada després de programar-les, és a dir, abans de realitzar el desplegament.

6.2 Proves unitàries de l'aplicació mòbil

Quant a la programació en Android, també s'ha seguit el mateix procediment, comprovant que totes les funcions implementades funcionessin correctament després de programar-les. Malgrat Android permet comprovar-ho a través de diversos emuladors, això alienta moltíssim el funcionament de l'ordinador i no permetia treballar de manera fluida, així que es va optar per l'opció de fer *debug* en un telèfon mòbil.

Pel que fa a la connexió amb el servidor, totes les proves s'han realitzat en local, és a dir, introduint la IP del ordinador utilitzat per fer-les, el port (fixat) i la ruta corresponent.

6.3 Proves funcionals

Les proves funcionals són les últimes que s'han dut a terme i s'encarreguen de comprovar el correcte funcionament del programa un cop està finalitzat i tot integrat. Per això s'han fet després de realitzar el desplegament de l'aplicació i amb el servidor web extern ja configurat i funcionant.

Malgrat no s'ha seguit una metodologia sistemàtica i automatitzada basada en tests a l'hora de realitzar aquestes proves, s'ha comprovat que el flux de l'app és el correcte tant si es passen totes les pantalles seguides com si s'abandona en qualsevol de les fases possibles i es torna a reprendre, que totes les advertències d'error o avis apareixen quan ho han de fer, que totes els conceptes i imatges funcionen correctament, que no hi ha problemes gràfics, que no hi ha problemes en tirar enrere, etc.

7 Validació preliminar

S'ha dut a terme un procés de validació preliminar de l'aplicació per tal d'avaluar-ne la usabilitat i l'acceptació. S'empra el terme validació preliminar perquè es tracta d'una mostra de reclutament petita (de 12 persones) i el procés de reclutament s'ha realitzat a través de gent propera. Caldria una mostra més gran i aleatòria per poder considerar els resultats (del tot) significatius. Els resultats són, per tant, orientatius i encarats a conèixer la reacció i avaluació que fan de l'aplicació alguns usuaris potencials, així com identificar possibles errors i millores.

Es vol comprovar si l'aplicació ha complert els seus objectius, és a dir, motivar els usuaris perquè acabin de respondre totes les preguntes i amenitzar el procés. És per això que s'han presentat les 2 versions de l'aplicació, la versió final gamificada presentada al llarg d'aquest treball i la versió no gamificada. La versió bàsica o no gamificada presenta un menú principal amb totes les preguntes del qüestionari IPSAQ, des del qual es dona l'opció d'escollir quina es vol respondre en cada moment, o també d'anar responnent-les totes seguides fent clic al botó de següent.

S'ha presentat cadascuna de les versions a 6 persones diferents per tal de poder comparar els resultats. No s'ha considerat oportú presentar les dues aplicacions a la mateixa persona perquè la reiteració en fer aquest llarg test provoca una degradació de l'experiència.

S'ha preparat una enquesta per avaluar la usabilitat de l'aplicació i la satisfacció de l'usuari. S'han agafat preguntes del SUS (*System Usability Scale*), algunes de les quals s'han modificat per adaptar-les a l'aplicació IPSApp en particular, i se n'han afegit d'altres relacionades amb l'estètica i la gamificació.

7.1 Preguntes de l'enquesta

Totes les preguntes de l'enquesta es mostren a continuació.

IPSApp

IPSApp és una aplicació mòbil que permet respondre el qüestionari d'estil atribucional IPSAQ de manera lúdica. Aquest qüestionari forma part d'un treball de fi de grau i avalua la teva experiència com a usuari de l'aplicació mòbil. Moltes gràcies per participar!

Versió de l'aplicació utilitzada

☐ Bàsica

☐ Gamificada

M'ha agradat utilitzar l'aplicació

	1	2	3	4	5	
Molt en desacord	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Molt d'acord

M'ha semblat una aplicació innecessàriament complexa

	1	2	3	4	5	
Molt en desacord	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Molt d'acord

M'ha semblat una aplicació fàcil d'utilitzar

	1	2	3	4	5	
Molt en desacord	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Molt d'acord

He trobat l'aplicació inconsistent

	1	2	3	4	5	
Molt en desacord	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Molt d'acord

He trobat l'aplicació incòmoda d'utilitzar

	1	2	3	4	5	
Molt en desacord	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Molt d'acord

M'he avorrit utilitzant l'aplicació

	1	2	3	4	5	
Molt en desacord	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Molt d'acord

M'agrada el disseny de l'aplicació

	1	2	3	4	5	
Molt en desacord	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Molt d'acord

He entès bé les preguntes del qüestionari

	1	2	3	4	5	
Molt en desacord	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Molt d'acord

El joc ha millorat la meua experiència com a usuari (si aplica)

	1	2	3	4	5	
Molt en desacord	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Molt d'acord

El joc m'ha motivat a acabar de respondre (si aplica)

	1	2	3	4	5	
Molt en desacord	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Molt d'acord

Algun suggeriment de millora o comentari?

Tu respuesta

7.2 Anàlisi dels resultats

Es recorda que la mostra consisteix de 12 persones, de les quals el 50% han provat la versió no gamificada i l'altre 50% la gamificada

Versió de l'aplicació utilitzada

12 respuestas

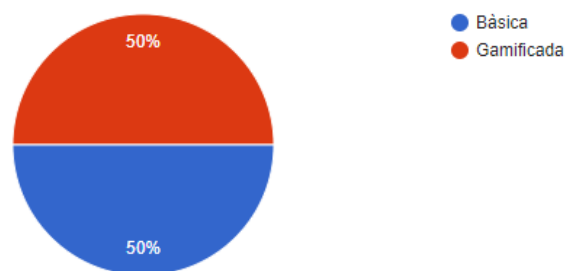


Figura 24. Percentatge d'ús de cada versió

En primer lloc, s'ha realitzat una comparació de la mitjana obtinguda, de l'1 al 5, per a cadascuna de les preguntes a respondre per a les 2 versions.

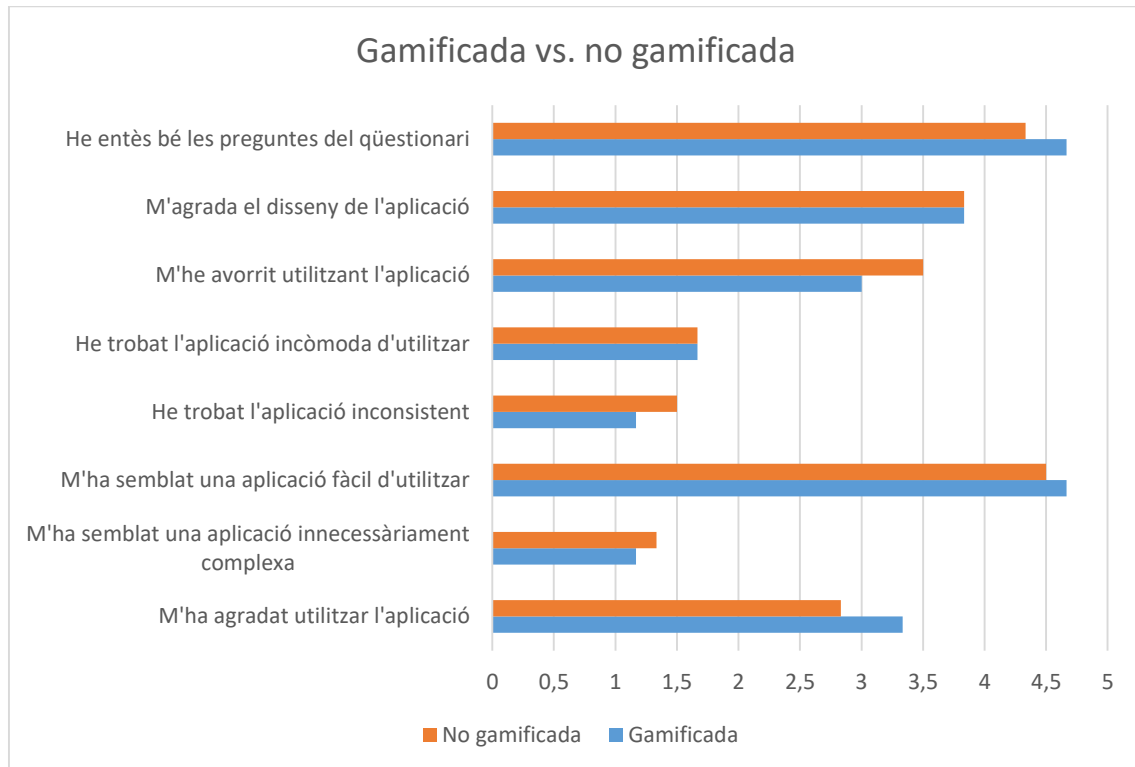


Figura 25. Aplicació gamificada vs. no gamificada

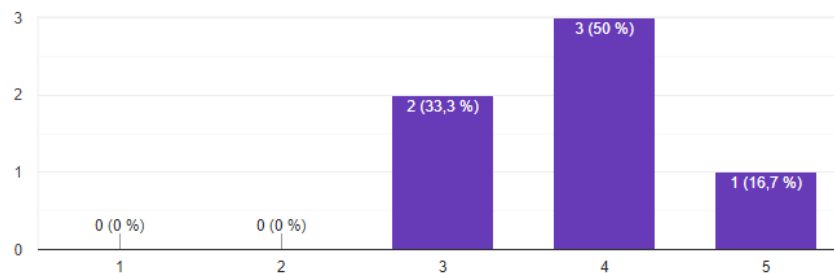
Els resultats d'aquesta enquesta corroboren que la majoria de participants troben el procés de respondre el qüestionari avorrit. Malgrat això, aquest efecte s'atenua en respondre a la versió gamificada. A més, els participants valoren positivament el disseny, consistència i la facilitat i comoditat d'ús de l'aplicació. D'altra banda, els usuaris no han tingut cap problema a l'hora d'entendre les preguntes. Caldria veure si això segueix essent de la mateixa manera en avaluar l'aplicació amb pacients.

En segon lloc, es mostren els resultats de les 2 preguntes que només fan referència a la versió gamificada.

El joc ha millorat la meva experiència com a usuari (si aplica)



6 respuestas



El joc m'ha motivat a acabar de respondre (si aplica)



6 respuestas

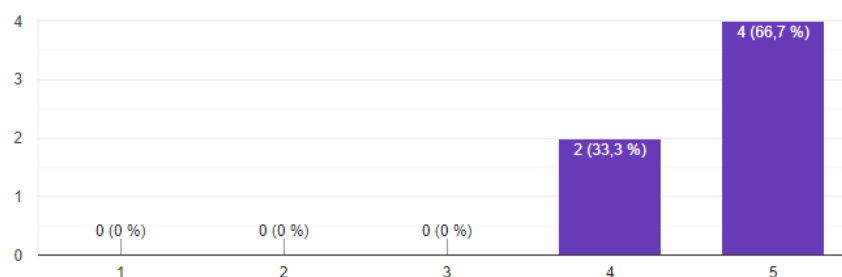


Figura 26. Gràfiques de les preguntes relatives al joc

S'observa que els usuaris de la versió gamificada indiquen que el joc els ha motivat molt a l'hora d'acabar el qüestionari i que ha millorat la seva experiència com a usuaris.

Algun suggeriment de millora o comentari?

6 respuestas

Que el programa m avisii quan dono una resposta al puzze de 2 paraules
Les preguntes són molt repetitives i s'ha d'escriure massa
Afegir una barra de progressió
Una mica monòton.
Trobo que el joc és la gràcia.
Alguna imatge difícil d'encertar

Figura 27. Comentaris i suggeriments rebuts

Un dels comentaris obtinguts al final de l'enquesta ha estat que les preguntes del qüestionari són repetitives. Això, però, no està en mans de l'estudiant ja que l'IPSAQ és un qüestionari psicològic preestablert. A més, un usuari de la versió bàsica indica que el

qüestionari és una mica monòton, la qual cosa és un dels principals motius pels quals s'ha afegit la gamificació. Algú ha suggerit que aparegués un missatge d'error en introduir una resposta amb més d'una paraula en el joc. Cal comentar que en el quadre de diàleg en què s'introdueix la resposta s'indica que aquesta correspon a sola paraula. Malgrat això, es podria afegir el missatge d'alerta com a possible millora per tal que l'usuari s'adonés si no ha vist la instrucció. També s'ha rebut el suggeriment d'afegir una barra de progressió. Es creu que aquesta és una molt bona idea com a futura millora ja que incrementaria el sentit de progrés i la motivació de l'usuari per acabar. Finalment, s'ha rebut la crítica que alguna imatge és difícil d'encertar, la qual cosa està feta intencionadament per afegir dificultat al joc, però es podria canviar fàcilment, i algú ha comentat que el joc és la gràcia de l'aplicació.

Cal recalcar, per últim, que els resultats d'aquesta validació són només orientatius. L'aplicació hauria de validar-se amb una mostra bastant més gran i aleatòria de pacients reals per poder considerar els resultats significatius.

8 Planificació i anàlisi econòmica

8.1 Planificació i realització

En aquest apartat es mostra una comparació de la planificació inicial del projecte amb la seva realització final. La càrrega de treball en hores es desglossa en la taula següent.

Activitat	Durada programada [h]	Durada real [h]
Recollir requeriments del client	10	5
Fer anàlisi funcional	25	25
Dissenyar arquitectura	30	25
Adquirir coneixements previs pel servidor	20	40
Programar servidor	50	80
Proves unitàries servidor	20	20
Adquirir coneixements previs per l'app mòbil	30	40
Programar aplicació mòbil	130	130
Proves unitàries app mòbil	20	20
Posada en marxa	15	10
Proves funcionals	10	10
Validació preliminar	10	20
Redactar documentació	50	50
	420	475

Cal destacar que l'apartat de recollir requeriments del client s'ha realitzat a través de les diferents reunions amb la tutora.

La inversió en hores del projecte ha estat major del que s'esperava fonamentalment a

causa d'una manca de coneixements previs. Així, la dedicació per adquirir els coneixements necessaris per a la realització del projecte, especialment en relació amb la programació del servidor i l'aprenentatge de Java, ha estat molt superior al que s'havia planificat, concretament en 30 hores. A més, a l'hora de programar el servidor, s'ha vist que la implementació era més complexa del que s'esperava a causa de la quantitat d'eines a utilitzar, pel que també s'ha excedit la càrrega de treball esperada en 30 h.

A continuació es mostra la distribució del treball en un diagrama de Gant:

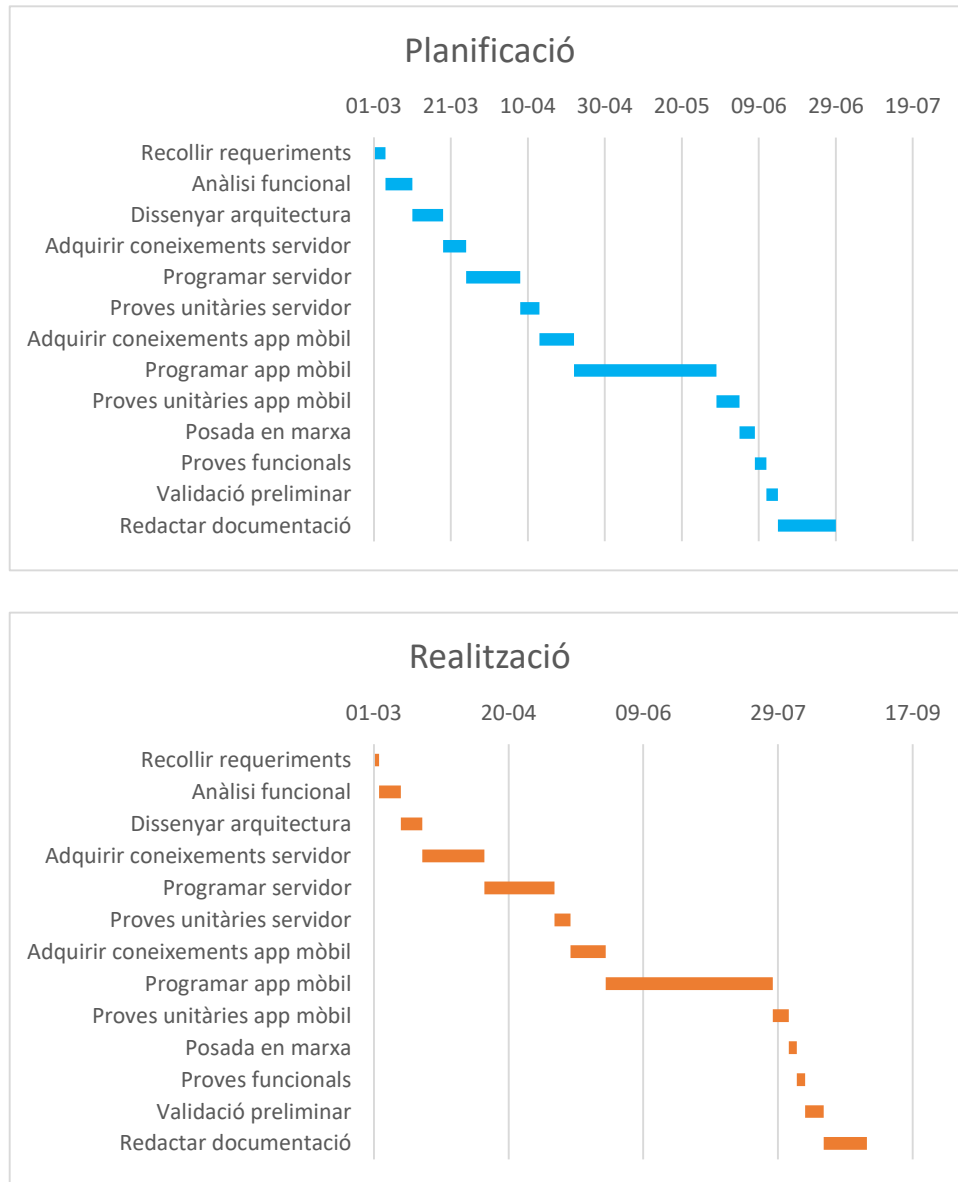


Figura 28. Diagrames de Gant de la planificació (a dalt) i realització (a baix) del treball

Cal destacar que el present treball es va registrar en la matrícula extraordinària, a finals de febrer. Es per això que es marca l'1 de març com a data d'inici. En un principi, es pretenia presentar el treball en la convocatòria de juny, i es va distribuir la càrrega de treball

equitativament entre el temps disponible. En la pràctica, es va intentar seguir la planificació però, en arribar l'època d'exàmens parcials, es va veure que la dedicació planejada era inassolible i es va començar a endarrerir el projecte. En arribar el més de maig, ja es va veure que amb els exàmens finals de per mig i havent invertit més temps del planificat en adquirir coneixements previs no es podia acabar el projecte a temps i es va decidir deixar-lo per la convocatòria del setembre. Així, a causa del fet que les hores totals del projecte van acabar sent superiors a les planificades i que durant certs períodes la dedicació no va poder ser suficient, el projecte va requerir 2 mesos més del previst.

Cal mencionar que en cas de realitzar aquest projecte en una empresa, la distribució del treball seria molt diferent perquè moltes de les tasques es durien a terme de manera simultània per diversos treballadors. En aquest cas, però, cap de les tasques es realitzen simultàniament ja que tota la feina està feta per una sola persona.

8.2 Pressupost

Per calcular el pressupost del projecte, s'ha tingut en compte el cost que tindria la feina realitzada en una empresa real, d'acord al sou dels treballadors necessaris per realitzar les diferents tasques que componen el projecte i el nombre d'hores assignades a cadascun d'ells. Es té en compte que, com que el material i software necessari per desenvolupar el projecte és convencional i no requereix cap llicència ni aparell amb característiques especials, el sou dels treballadors ja inclou el cost d'amortització del material de hardware empleat i de l'energia consumida, ja que aquesta metodologia representa la manera d'actuar de les empreses a l'hora de realitzar el pressupost d'un projecte. Tot i això, es presentaran també els costos mencionats a títol purament informatiu.

8.2.1 Recursos humans

La realització del projecte en una empresa requereix d'almenys 4 treballadors:

- Un cap de projecte i analista: s'encarrega de gestionar el projecte, mantenir el contacte amb el client i fer una anàlisi funcional.
- Un programador sènior i arquitecte: s'encarrega d'escollir la tecnologia usada en cada part del programa i de programar les parts més difícils.
- Dos programadors juniors: un que s'ocupa de programar el servidor i l'altre de l'app mòbil.

Habitualment els rols de cap de projecte i analista els exerceixen 2 persones diferents, igual que els de programador sènior i arquitecte. Tot i això, aquí s'assumeix el cas d'una empresa petita en què aquests rols es podrien agrupar i assignar a la mateixa persona.

Malgrat en aquest projecte tots els rols els ha assumit la mateixa persona, es diferencien els 4 rols per tal de calcular el cost aproximat de la feina realitzada. Es presenten les

tasques assumides i el nombre d'hores invertides per cada treballador en la taula següent:

Rol	Activitat	Hores	Hores totals
Cap de projecte i analista	Recollir requeriments	5	110
	Anàlisi funcional	25	
	Proves funcionals	10	
	Validació preliminar	20	
	Redactar documentació	50	
Programador sènior i arquitecte	Dissenyar arquitectura	25	35
	Posada en marxa	10	
Programador junior (web)	Coneixements previs servidor	40	140
	Programar servidor	80	
	Proves unitàries	20	
Programador junior (Android)	Coneixements previs app mòbil	40	190
	Programar app mòbil	130	
	Proves unitàries	20	
Total			475

A continuació el presenta el cost total dels recursos humans diferenciats segons treballador. El cost del treballador s'ha calculat tenint en compte que treballa 1800 h/any, com marca el conveni que se sol aplicar, i afegint al salari que cobra un 33% extra que l'empresa paga a la seguretat social.

Quant als salaris, s'ha consultat un treballador de Lleida d'una empresa anomenada Indra i, en els casos d'un treballador que assumeix 2 rols, s'ha fet una mitjana aproximada d'ambdós.

			Hores de feina	Cost [€/h]	Total [€]
Mà d'obra		Cap de projecte i analista	110	26,6	2926
		Programador sènior i arquitecte	35	24,38	853,3
		Enginyer junior app web	140	13,3	1862
		Enginyer junior app mòbil	190	13,3	2527

8.2.1.1 Hardware

El cost d'amortització del material utilitzat per desenvolupar el projecte es mostra en la següent taula.

Producte	Preu [€]	Vida útil [anys]	Temps d'ús [mesos]	Cost d'amortització
Ordinador portàtil	450	6	6 mesos	37.5€
Mòbil	250	3	6 mesos	41.6€

Es negligeix el cost d'amortització del router a causa de les seves múltiples utilitats i la seva llarga vida útil.

Els càlculs s'han fet amb la fórmula següent:

$$\text{Cost amortització} = \text{Preu} \cdot \frac{\text{Temps d'ús}}{\text{Vida útil}}$$

8.2.1.2 Software

El cost dels diferents softwares instal·lats es mostren en la taula següent:

Software	Cost [€]
Windows	0*
Visual Studio Community	0
Android Studio	0
MySQL	0
SQLite	0
Google Chrome	0

*Windows té un cost adicional si es vol instal·lar a part, però no es té en compte ja que es trobava instal·lat amb l'ordinador i, per tant, el seu cost no s'ha pagat i, en tot cas, estaria inclòs dins del cost d'amortització de l'ordinador.

Es pot observar que el cost és de 0 € ja que tots els softwares utilitzats (excepte Windows) són de codi obert.

8.2.1.3 Costos indirectes

Els costos indirectes d'aquest projecte es constitueixen fonamentalment pel cost de l'energia elèctrica dels aparells utilitzats. Els costos associat al consum del router i a l'accés a internet no es tenen en compte ja que ambdós conceptes estan associats a múltiples usos no relacionats amb el present projecte i el nombre d'hores diàries en que se'n fa ús per aquest propòsit no és molt alt, motiu pel qual es consideren negligibles.

Així, tenint en compte el preu mitjà de l'electricitat a Espanya (uns 0,14 €/kWh), el temps d'ús de cada aparell i el seu consum elèctric, es calcula el cost mitjà del consum elèctric d'aquest projecte a través de la fórmula següent:

$$\text{Cost de consum elèctric[€]} = \text{consum elèctric[kW]} \cdot \text{temps d'ús[h]} \cdot \text{cost electricitat[€/KWh]}$$

S'assumeix que l'ordinador consumeix 50 W i s'ha utilitzat en totes les tasques realitzades, és a dir, 475 h. Pel mòbil, en canvi, se suposa un consum de 15 W i que només s'ha utilitzat durant les proves unitàries de l'app mòbil, les proves funcionals i les de validació, és a dir, 50 h.

Aparell	Consum [kWh]	Hores d'ús [h]	Cost total [€]
Ordinador	0,4	475	3,56
Mòbil	3,52	50	0,11
Total			3,67

8.2.2 Llicències

A més del cost del software emprat per desenvolupar l'aplicació, cal veure si els diferents mòduls o extensions instal·lats addicionalment en el programa són d'ús i comercialització lliure o no, per tal de conèixer si caldria sumar un cert cost en cas de voler utilitzar-lo per a fins comercials o de distribució. A continuació pot trobar-se una taula de les llicències dels diferents softwares dels quals fa ús el programa.

Software	Llicència
Python	PSFL
Flask	BSD
Flask-Security	MIT
Flask-Mail	BSD
Flask-Debug	MIT
Flask-WTF	BSD
Flask-BabelEx	BSD
Bootstrap-Flask	MIT
PyJWT	MIT
MySQLClient	GPL
Bcrypt	Apache 2.0
SQLite	<i>Public Domain</i>
Android	Apache 2.0

Les llicències MIT, BSD, PSFL i Apache 2.0 són lliures i permissives. Permeten l'ús, distribució o comercialització del software amb la condició que es preservi una nota amb el copyright i la llicència en totes les còpies del programa. *Public Domain* fa referència a que l'obra no està protegida per propietat intel·lectual, és de domini públic. L'única llicència més restrictiva és la GPL, ja que requereix que el codi font estigui disponible sota la mateixa llicència en totes les còpies distribuïdes. D'aquesta manera, complint les condicions de la GPL, si es volgués comercialitzar el programa no caldria afegir cap cost addicional. Cal tenir present, però, que en cas de no disposar d'un compte de desenvolupador a Google Play, caldria crear-ne un per tal de publicar l'app, la qual cosa representaria un cost de 25\$ a pagar un sol cop. Aquests costos no se sumen al projecte ja que no es té intencions properes de publicar l'app.

Per a la programació de l'app mòbil només s'han utilitzat extensions natives del propi Android. A més, les imatges són totes lliures de drets.

8.2.3 Total

Els costos totals es desglossen en la taula següent:

Concepte		Cost [€]
Recursos humans	Cap de projecte i analista	2926
	Programador sènior i arquitecte	853,3
	Enginyer junior app web	1862
	Enginyer junior app mòbil	2527
Llicències		0
Total		8168,3

El cost total del projecte ascendeix a 8168,3 €.

9 Impacte ambiental i social

L'impacte ambiental del present projecte és molt baix, ja que ha estat desenvolupat en la seva totalitat virtualment. Així, el cost ambiental només està associat a l'amortització dels materials usats per a la construcció de l'ordinador utilitzar per programar i el mòbil emprat per fer les proves, a més de l'energia consumida per aquests 2 aparells, i la de les comunicacions. Malgrat això, en no haver adquirit cap material nou pel projecte, sinó haver utilitzat el ja disposat per l'usuari, els costos ambientals associats a la seva amortització es podrien pràcticament negligir.

L'impacte social del projecte, en ser implementat per al seu ús en el món de la neuropsicologia, és molt positiu, ja que proporciona una nova eina amb la qual realitzar l'avaluació psicològica o psiquiàtrica de pacients. Aquesta eina no només simplifica la tasca pels usuaris, ja que poden realitzar els qüestionaris quan vulguin i fins i tot en diverses fases, sinó que els motiva i els proporciona entreteniment, en haver implementat la part de gamificació. Nogensmenys, la tasca de recull, anàlisi i seguiment dels resultats per part del psicòleg es facilita en gran mesura.

Conclusions

Un cop finalitzat el disseny, implementació i validació preliminar de l'aplicació client-servidor IPSApp, es pot concloure que s'han assolit en gran mesura els objectius del projecte. S'ha obtingut una aplicació funcional, amb una estètica senzilla però cuidada i coherent, que suporta 3 llengües i que aporta beneficis en el món de la psicologia per les dues bandes: la del psicòleg, que pot realitzar la gestió i seguiment dels pacients de manera més fàcil, i la del pacient, que pot respondre al qüestionari de manera més còmoda, lliure i entretinguda.

Malgrat la creació d'una aplicació d'aquestes característiques no és de gran dificultat tecnològica en el món professional, ha suposat un gran repte per l'estudiant en haver d'aprendre un nou llenguatge de programació (Java) i les tecnologies necessàries per crear un servidor i realitzar les connexions amb l'aplicació. El gran repte en el món de la psicologia esdevé implementar elements lúdics de manera que el qüestionari resulti més amè i es motivï el pacient per acabar-lo, sense distreure'l en excés per tal d'evitar que respongui les preguntes sense haver-les entès o pensat. Els resultats de la validació preliminar semblen indicar que s'ha aconseguit, almenys en certa mesura, el propòsit establert, a través d'un joc senzill entre preguntes que no requereix molt d'esforç o dedicació i que té un fil conductor que conclou en finalitzar el qüestionari.

Les assignatures del grau han estat de gran ajuda en la realització d'aquest projecte, en haver proporcionat les bases de Python i Flask necessàries per poder aprofundir més en l'ampli món de les aplicacions.

Finalment, cal destacar les possibilitats de l'aplicació de cara al futur. El joc és molt fàcilment escalable, a través de la incorporació de més conceptes i imatges associades a cadascun. També es molt senzill canviar el nombre de peces que conté cada imatge per tal d'ajustar la dificultat del joc. Es podria també millorar l'estètica de l'aplicació i afegir més elements lúdics, com un avatar, una barra de progressió, una historieta al llarg del qüestionari o fins i tot vinyetes que ajudessin a entendre les preguntes. A més, es poden canviar de forma molt senzilla les preguntes del qüestionari per tal d'adaptar l'aplicació al sistema d'avaluació utilitzat en cada moment.

El projecte, per tant, ha assolit els seus objectius i ha resultat molt beneficiós, tant per l'estudiant, que ha adquirit uns coneixements molt valuosos pel seu futur professional, com a nivell social, representant una petita contribució en el camp de la psicologia.

Agraïments

A la meva família, per la seva ajuda i implicació en aquest projecte. No hauria estat possible sense ells.

A la meva tutora, per la seva dedicació i orientació.

Bibliografia

Referències bibliogràfiques

- [1] ARNE HOLST, *Global mobile OS market share in sales to end users from 1st quarter 2009 to 2nd quarter 2018*, 2019
- [2] PETER KINDERMAN AND RICHARD P. BENTALL. *Person. individ. Diff.* Vol. 20(2), 1996, p. 261-264
- [3] Documentació de Flask. Disponible a: <https://flask.palletsprojects.com/en/1.1.x/>
- [4] Documentació de Bootstrap. Disponible a <https://getbootstrap.com/docs/4.3/getting-started/introduction/>
- [5] DRAGOS HOLBAN, *How to Build a Jigsaw Puzzle Android Game*, 2018. Disponible a: <https://dragosholban.com/2018/03/09/how-to-build-a-jigsaw-puzzle-android-game/>
- [6] MDB, *Email templates, Bootstrap responsive email templates*. Disponible a: <https://mdbootstrap.com/plugins/jquery/email-templates/>

Bibliografia complementària

- YOUTUBE, freeCodeCamp.org, *Learn Java 8 - Full Tutorial for Beginners*, 2018. Disponible a <https://youtube.com/watch?v=grEKMHGYYns>
- YOUTUBE, Pretty Printed, *Creating a RESTful API in Flask With JSON Web Token Authentication and Flask-SQLAlchemy*, 2017. Disponible a: <https://www.youtube.com/watch?v=WxGBoY5iNXY>
- IRSHAD KUMAIL *Android Volley Example- HTTP GET,POST with Custom Headers*, 2017. Disponible a: <https://androidclarified.com/android-volley-example/>
- YOUTUBE, Coding in Flow, *Custom Dialog + Sending Information to Activity - Android Studio Tutorial*, 2017. Disponible a <https://www.youtube.com/watch?v=ARezg1D9Zd0>.
- YOUTUBE, Coding in Flow, *Options Menu with Sub Items - Android Studio Tutorial*, 2017. Disponible a <https://www.youtube.com/watch?v=ARezg1D9Zd0>.
- GeeksforGeeks. *Android | Creating a Splash Screen*. Disponible a: <https://abhiandroid.com/programming/splashscreen>
- RAVI TAMADA, *Android SQLite Database Tutorial*, 2018. Disponible a:

<https://www.androidhive.info/2011/11/android-sqlite-database-tutorial/>

RAVI TAMADA, Android User Session Management using Shared Preferences, 2017.

Disponible a: <https://www.androidhive.info/2012/08/android-session-management-using-shared-preferences/>

Stack Overflow. Disponible a: <https://stackoverflow.com/>